

## Studie «Terminal Services»

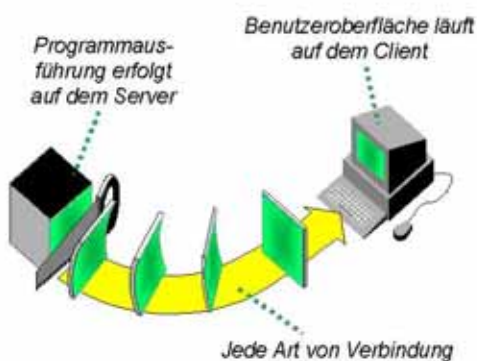
### Ausgangslage

Im Rahmen der Projektleitung Ausbildungsinformatik wünscht RPL Ben eine einführende Studie zu Vor- und Nachteilen der Terminal/Server Architektur. Der Einsatz von Thin Clients ist nur teilweise Bestandteil dieser Studie (siehe Grundlagen).

### Grundlagen

Versuchsauftrag der FA 16 zu Microsoft Terminal Server Edition und Citrix MetaFrame mit Microsoft Windows 2000 Server (alle Versionen). Recherchen im Internet betreffend Leistungsumfang, Systemverhalten und Lizenzbedingungen. Demo bei DELEC zu Server-based Computing mit Citrix MetaFrame und eigene Erfahrungen mit VideoFrame anhand der InternetExpo 2000.

### Windows NT Terminal Server



### RDP von Microsoft ICA von Citrix

#### Was sind Terminal Services ?

Der Unterschied zwischen Terminal Services und einer traditionellen Mainframe-Umgebung besteht darin, dass die sogenannten «dummen Terminals» des Mainframe-Computers nur über text- oder zeichen-basierende Ein-/Ausgabe verfügen. Ein Windows Terminal Server stellt mit Hilfe der Terminal Services gleichzeitig mehrere Client Sessions zur Verfügung, auf welche sich die Benutzer von einem Terminal einloggen, und auf dem Server jegliche 16- und 32-bit Applikationen betreiben können. Die Anwendungen laufen also effektiv auf dem Server ab, nur die Bild- und Toninformationen kommen zum Client.

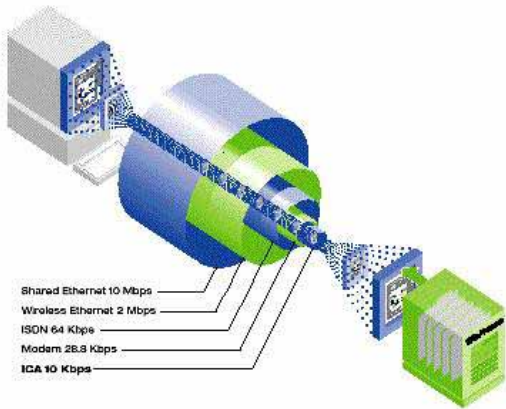
#### Was ist ein ThinClient ?

Ein Terminal Services Client ist ein so genanntes ThinClient-Gerät, welches über ein integriertes Betriebssystem verfügt. Dieser stellt die Verbindung zum Terminal Server her. Der Terminal Server Emulator ist die Client Software, welche auf einem Windows-(16- und 32-bit), Macintosh- oder UNIX-Client installiert wird, um sich auf den Terminal Server einloggen zu können und den Verbindungsaufbau zu starten.

#### Wie funktioniert die Kommunikation ?

Ein wichtiger Bestandteil des Terminal Servers, der die Kommunikation zwischen Server und Client ermöglicht, ist das Protokoll. RDP von Microsoft bietet 64'000 Transportkanäle für Serial Device Communication und Presentation Data sowie verschlüsselte Mouse- und Keyboard-Daten.

Das Design von RDP ermöglicht Verbindungen über verschiedenste Protokoll-Typen wie zum Beispiel ISDN, IPX, Netbios usw. Die aktuelle Version unterstützt jedoch nur TCP/IP. Analog dem OSI-Layer-Prinzip werden Daten von Applikationen und Services durch die verschiedenen Protokoll-Stacks in Pakete und Rahmen aufgeteilt, verschlüsselt und gepackt, und schliesslich adressiert und über das LAN/WAN zum Client geschickt. Die Antwort-Daten des Clients werden in umgekehrter Reihenfolge verarbeitet und wieder ausgepackt, so dass sie im Server wieder der Applikation zur Verfügung stehen.



**Independent Computing Architecture (ICA)**

Ein von der Firma CITRIX entwickeltes Client-Server-Protokoll. Es unterstützt direkte asynchrone Kommunikation und neben TCP/IP auch IPX/SPX, NetBEUI, ISDN, Frame Relay und ATM als Transportprotokolle.

**WinFrame/MetaFrame**

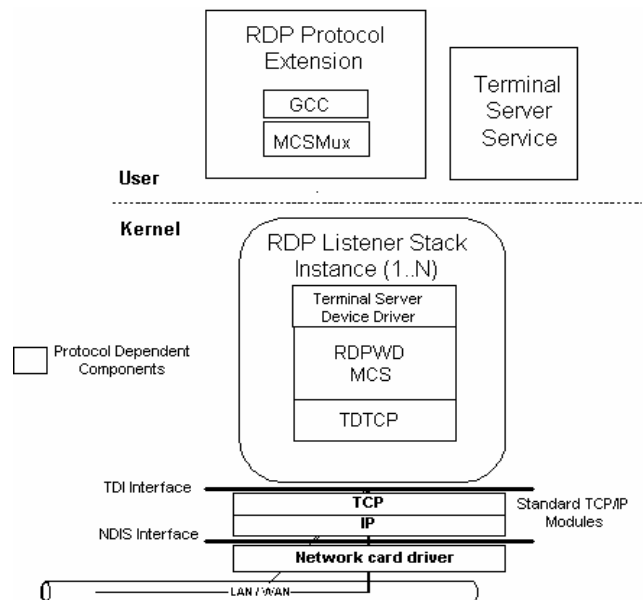
ebenfalls zwei Produkte von CITRIX. WinFrame ist eine Multi-User Version von Windows NT 3.51. Mit MetaFrame kann eine Windows NT 4.0 TSE Server Farm durch mehrere Features erweitert werden. MetaFrame unterstützt fast alle möglichen Client Betriebssysteme und bietet Multimedia- und Ressourcenmanagement-Funktionen .

Hier die wichtigsten Vorgänge:

**Terminal Server Initialisierung**

Beim Laden des Kerns eines Windows NT Terminal Servers wird der Terminal Server Service (termsrv.exe) gestartet und dieser wartet dann auf Session Connections. Jede dieser Connections erhält eine SessionID, welche auch gleich den damit erzeugten Prozessen angehängt wird, um sie zu identifizieren. Die Console Session (Terminal Server-Keyoard, -Maus und -Video) wird immer als erstes geladen und mit der SessionID0 versehen. Danach ruft der Windows NT Session Manger (SMSS.EXE) zwei neue Client Sessions auf, welche auf Client Connections warten. Danach wird der Client Server Run-Time Prozess (CSRSS.EXE), der Winlogon Prozess (WINLOGON.EXE) und das Kernel Modul Win32k.sys geladen.

Durch das spezielle Design des Terminal Server Kerns werden pro Session immer nur diejenigen Teile des Codes neu in den Speicher geladen, welche sich zu den bereits bestehenden Sessions unterscheiden, d. h. gemeinsam genutzter Code wird also nur einmal geladen.



**Wie funktioniert die Client Connection ?**

Ein sogenannter Terminal Server Listener erkennt eine Client-Anfrage und initialisiert eine neue Instanz des RDP-Stacks, welcher er dann die eingehende Connection übergibt um sogleich wieder den TCP-Port nach neuen Anfragen abzuhören.

Bei jeder neuen Session wird als erstes der Verschlüsselungsgrad (vom Client bestimmt) ermittelt, wobei der Terminal Server deren drei anbietet: low, medium und high.

Bei allen Clients werden Speicher zum Cachen von Bitmaps (Icons, Toolbars, Cursors) reserviert.

Der Terminal Server besitzt ebenfalls Buffer, um Bildschirm-Refreshes kontrolliert zu übermitteln (anstelle eines konstanten Bitstreams). Bei hoher Interaktivität der Benutzers (Keyboard- und Mausbewegungen) wird dieser Buffer ca. 20 mal pro Sekunde geflutet, bei ruhigem Interaktions -Verhalten geht diese Rate zurück auf 10 mal pro Sekunde. Die Belastung der Verbindung wird also so niedrig wie möglich gehalten.

Zum jetzigen Zeitpunkt, noch bevor überhaupt ein Anmeldefenster erscheint, werden die Lizenzbedingungen getestet und dem Client wird eine Client Access Lizenz vergeben.

Nachdem alle konfigurierbaren Parameter der Connection ausgehandelt sind, wird das Anmeldefenster angezeigt und der Benutzer kann Name und Passwort eingeben. Der Winlogon Prozess authentifiziert den Benutzer und übergibt den Namen und dessen Domain-Zugehörigkeit an den Terminal Server Service, welcher eine Liste mit Domain/Benutzernamen und SessionID führt.

### Starten einer Applikation

Der Code im Kernel- und User-Mode in einem Terminal Server wird wenn immer möglich von allen Prozessen gemeinsam genutzt, dazu belegt der Virtual Memory Manager (VMM) alle Pages mit einer „copy-on-write page protection“. Im Falle eines Schreib-Zugriffes werden die entsprechenden Pages vom VMM an einen anderen Ort im RAM kopiert und mit read/write-Berechtigung versehen.

Dies bedeutet, dass bei einer Win32-Applikation, wie zum Beispiel Microsoft Word, der grösste Teil des Codes nur einmal in den Speicher geladen wird, und dass nur spezifische lokale Buffer und Daten für jeden Benutzer individuell mit read/write-Berechtigung im RAM vorhanden sind. Dieses copy-on-write-Design macht den Terminal Server besonders effizient für den Betrieb als Applikations-Server für 32-bit Programme.

Weil beim Windows Terminal Server jede 16-bit Anwendung in ihrer eigenen Virtual DOS Machine (NTVDM) gefahren wird, kann der Speicher nicht gemeinsam genutzt werden. Das Übersetzen von 16-bit in 32-bit Befehle verbraucht noch zusätzlich Ressourcen, was sich mit einer Leistungseinbusse von bis zu 40% gegenüber einer vergleichbaren 32-bit Applikation bemerkbar macht.

### Beenden einer Applikation

Sobald ein Benutzer seine Verbindung unterbricht, werden alle „seine“ Prozesse und virtuellen Memory-Pages auf die Festplatte geschrieben. Beim nächsten Verbinden werden alle diese Benutzer-Daten wieder in den Speicher geladen und der Benutzer kann mit seiner gewohnten Umgebung weiter arbeiten. Eine spezielle Eigenschaft von RDP erlaubt sogar das automatische Anpassen der Bildschirmauflösung beim Reconnect von einem anderen Client. Das Abmelden ist typischerweise eine einfache Angelegenheit. Alle Prozesse mit der entsprechenden SessionID werden geschlossen und jeglicher Speicherplatz wird freigegeben. Natürlich bleiben mehrfach benutzte Applikationen solange im Speicher, bis sie der letzte Benutzer sie schliesst.

## Vorteile Terminal / Server

- Schnelle Installation von Anwendungen
- Die Einführung/Upgrade von Software ist speditiv für das gesamte Unternehmen möglich
- „schwimmende“ Migration nur zwischen den Servern nötig
- Kosteneinsparungen durch den Einsatz von ThinClients
- Drastische Reduktion der Unterhaltskosten (TCO)
- Schutz der zuvor getätigten Hardware-investitionen
- Teilweise einzige Möglichkeit für nicht-redundante Datenhaltung bei dezentralen Anwendern
- Lastausgleich und Redundanz über mehrere Server
- Zentraler Benutzer-Support durch Übernahme der Benutzersession
- TCP/IP, IPX, SPX, NetBEUI/NetBIOS, direkt asynchron werden unterstützt
- automatisierter Client-Software Update
- Spiegelung von Anwendersitzungen, d.h. ein Lehrer kann z.B. zwischen den Clients hin- und herschalten

## Lizenzbedingungen

### Microsoft

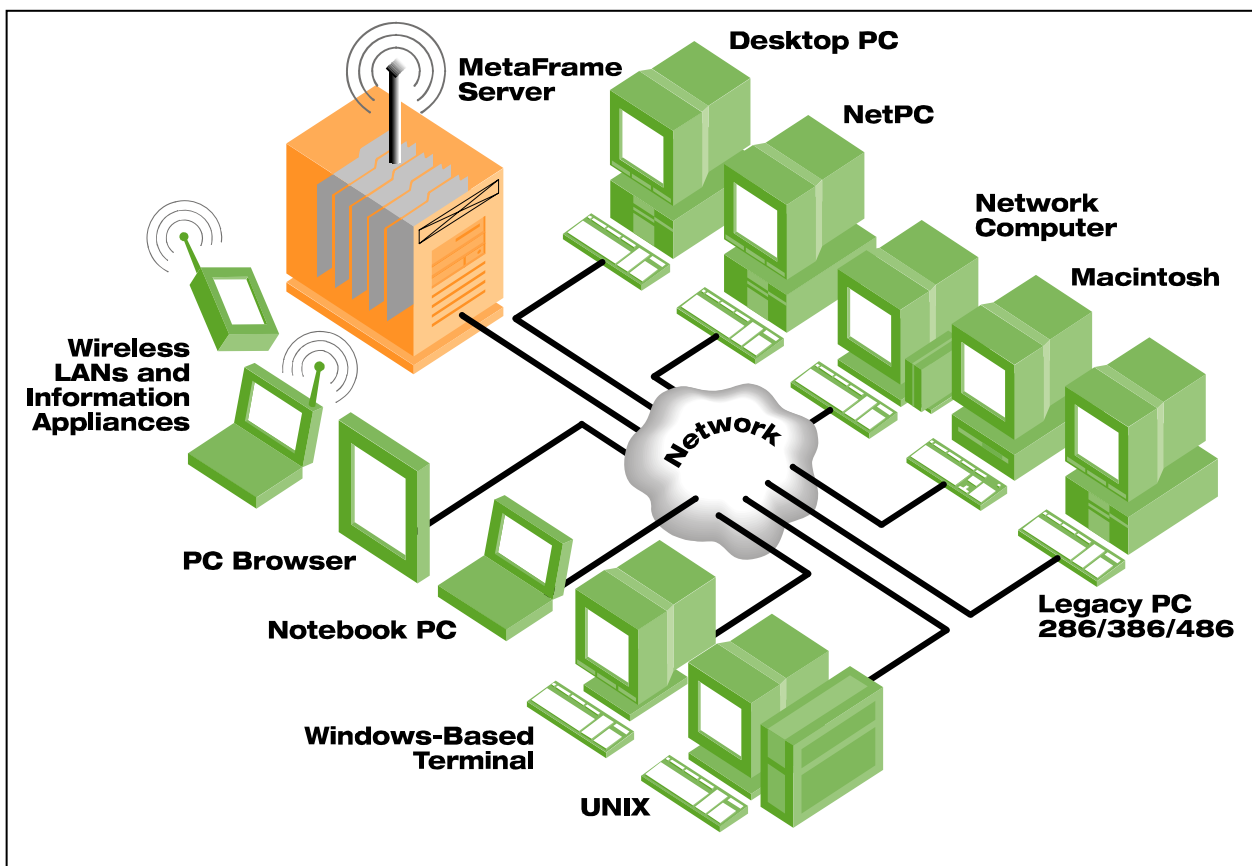
- Microsoft Win NT 4.0 Terminal Server  
multiuserfähiges Betriebssystem auf Basis  
Win NT 4.0 incl. 10 Client Access Licenses in e und d, Clientprotokoll ist RDP
- Microsoft Win NT CAL oder BackOffice CAL
- Microsoft Win NT 4.0 Workstation oder WTS CAL

- Microsoft WTS Internet Connector für max. 200 gleichzeitige anonymous User

### Citrix (WinFrame)

- WinFrame Enterprise  
Applicationsserver incl. Multiuserkernel NT3.51 für 15 gleichzeitige Benutzer in e, d, f, s incl. File- und Printservices integriert
- WinFrame for Workgroups  
Applicationsserver incl. Multiuserkernel NT3.51 für 5 gleichzeitige Benutzer in e, d, f, s ohne File- und Printservices, keine Erweiterung der Userzahl möglich
- User License Pack  
Erweiterung um 5, 10, 20 oder 50 weitere Benutzer
- ICA Security Option Pack  
Zusatzoption zur ICA-Verschlüsselung

## Thin-Clients im Überblick



ANHANG:

**Vergleich der beiden Protokolle**

Available Clients	RDP	ICA
Windows NT	YES	YES
Windows 95 / 98	YES	YES
Windows 3.11 (Windows for Workgroups)	YES	YES
Windows CE	YES <sup>(1)</sup>	YES
DOS		YES
Apple Macintosh		YES
Browser (MS Internet Explorer)		YES
Browser (Netscape Navigator)		YES
UNIX -- Solaris / Sparc		YES
UNIX -- Solaris / x86		YES
UNIX -- SunOS		YES
UNIX -- DEC		YES
UNIX -- HP/UX		YES
UNIX -- IBM		YES
UNIX -- SGI		YES
UNIX -- SCO		YES
Linux (RedHat, Caldera, SuSe, SlackWare)		YES
Java -- JDK 1.0 / 1.1		YES
DirectConnect		YES

Transport Protocols	RDP	ICA
TCP/IP	YES	YES
IPX/SPX		YES
NetBEUI		YES

Multimedia	RDP	ICA
System Sounds (speaker beeps)	YES	YES
16-bit Stereo		YES
Video		YES
Multimedia bandwidth control		YES

## Glossar

**Local Printer Support** – Basically states that a client can see a local attached printer as if it were on the server running the session.

**Session Shadowing** – Allows the I/O of a session to be distributed among several users on several clients. This way an administrator can shadow a user for various purposes (support, training, monitoring...).

**Client Redirection** – It is the ability for a client to copy, cut and paste between applications running locally and those running in a terminal session.

**Load Balancing Support** – Allows one application to serve different users on a server farm composed of different servers. This way the load generated by many users can be split.

**SpeedScreen** – Intelligent agent that reduces the transmission of frequently repainted screen, thus reducing the network traffic. Basically, it only repaints changing portion of the screen.

**Seamless Windows** – The applications running on a remote server can “seamlessly” be integrated into the local desktop.

**Business Recovery Client** – Automatically connects users to a backup Metaframe or Winframe server farm in case of session disconnection.

**Program Neighborhood** – Allows administrators to make applications available in a simpler way. An icon for the new application shows up regardless of the client configuration.

**Ready Connect Client** – Greatly simplifies and accelerates the mass deployment of clients via standard predefined clients.

**Auto Client Update** – Centralizes and schedules the download and installation of a client software on the client.

**Shadow Task Bar** – Supports shadowing of multiple users from a single desktop.

**Resource Management Services** – Provides real-time monitoring and management of application, user and system resources.

**Installation Management Services** – Packages applications once to publish them on all the server of the farms you may have.