

17 Transport von X.11-Anwendungen

Während bei Windows erst Ende des letzten Jahrhunderts und Anfang dieses Jahrhunderts die Server-Produkte Windows NT 4-Server-Terminal-Server-Edition, Windows 2000-Server und Windows 2003-Server durch Schützenhilfe von Citrix *transportfähig* wurden, war es in Linux/Unix schon Mitte der 1990er Jahre selbstverständlich, dass man von einem beliebigen Rechner aus eine Desktop-Sitzung auf einem anderen PC nutzen konnte. Das Massachusetts Institute of Technology (MIT) hatte schon damals auf der Basis der Vorgängerprojekte »V« und »W« das X-Windows-System als grafische Benutzeroberfläche für Unix-Systeme entwickelt.

Anwendungs-Server und das Benutzer-Endgerät kommunizieren dabei über das X.11-Protokoll. Dabei sind die Begriffe *Server* und *Client* anders als gewohnt definiert:

- Am Benutzer-Endgerät läuft ein X-Server, der Benutzereingaben (Maus, Tastatur) entgegennimmt und Ausgaben auf seinem Bildschirm darstellt.
- Anwendungen, X-Clients wie Textverarbeitung, Mail etc. nutzen diese grafische Oberfläche, indem sie von ihr die Benutzereingaben erhalten und ihr Daten zur Ausgabe auf deren Grafikbildschirm schicken.

17.1 Überblick

Da im Abschnitt 16.2 dieses Buches bereits beschrieben ist, wie man die Grafikausgaben eines X-Clients (Anwendungs-Servers) auf einen X-Server (PC als X-Terminal) zaubert, kann sich der Abschnitt 17.2 zu X.11 auf dessen Grundprinzipien und Schwächen beschränken, die den Bedarf an eleganteren Transportlösungen wie Nomachine NX und Tarantella Secure Global Desktop Enterprise Edition (SGD EE) erklären:

- Der Clou der Nomachine-Lösung (Abschnitt 17.3) ist sein intelligentes Caching der Dialoge zwischen dem X-Client und dem X-Server, verbunden mit Kompression und einer SSL-Verschlüsselung.
- SGD EE (Abschnitt 17.4) übersetzt den X.11- Verkehr in sein eigenes Adaptive Internet Protocol (AIP) und komprimiert/verschlüsselt den Datenverkehr abhängig von der zur Verfügung stehenden Bandbreite.

Durch die Verwendung eigener Protokolle bieten beide Anbieter eigene Clients zur Emulation ihrer Protokolle an und erreichen dadurch eine Entkopplung der beiden Schichten. Hier liegt der eigentliche Unterschied zwischen den Lösungen: Während die EE als »Applikations-Portal« bei Anwendern keine Konfiguration benötigt, konfiguriert man den Applikations-Server von NX am Client.

Probefahrten mit Nomachine NX oder Tarantella EE können Sie über die Adresse <http://www.nomachine.com/testdrive.php> bzw. <http://www.tarantella.com/demo/> starten. Hier haben Sie die Gelegenheit, vorab einen Eindruck über die in den folgenden Kapiteln beschriebene Software zu erlangen.

17.2 X.11-Programme im Remote-Betrieb

Theoretisch ist es X-Programmen völlig egal, ob sie ihre Daten lokal oder remote über ein Netz übertragen, ob der X-Server also auf demselben PC läuft oder auf einem weit entfernten. Bei X-Programmen tauschen der X-Client und X-Server über das Protokoll X.11 Fragen und Antworten miteinander aus und warten dabei immer höflich auf alle Antworten auf jede Frage. So eine Frage mit Antwort heißt auch *round trip*.

Ein- und dieselbe Ausgabe kann viele oder wenige Dialoge erfordern. So kann der X-Client für ein animiert aufklappendes Menü

- den X-Server jeweils eine Pixelzeile schreiben lassen und ihn dann fragen, ob er das auch wirklich geschrieben hat und dann auf die positive Antwort warten, bevor er ihm die nächste Pixelzeile schickt oder
- den X-Server pro 500stel Sekunde oder in noch kürzeren Abständen eine Zeile schreiben lassen und nach dem Aufbau des gesamten Menüs fragen, ob er damit fertig ist. Das erfordert nur einen *round trip*.

Führt man diese beiden Menü-Varianten lokal auf einem PC aus, wird man kaum Unterschiede feststellen, da hier so genannte Unix Domain Sockets die Wege zwischen X-Client und X-Server überbrücken. Über ein schmalbandiges WAN werden Benutzer hingegen bei der ersten Version unnötig lange auf die Übertragung der vielen Frage-Antwort-Spielchen zwischen X-Client und X-Server warten müssen. Testen Programmierer ihre Anwendungen nur lokal, merken sie nicht, wenn ein X-Programm viele solcher *round trips* enthält. Dabei sind Programmierer daran nicht allein »Schuld«, da viele dieser *round trips* eine Folge der Verwendung von Programmbibliotheken sind, die für lokale Anwendungen optimiert sind und sich nicht darum scheren, wenn bei Ausführung über ein WAN der Mauszeiger zuckelt und die Animationen ruckelt. Durch das sinnlose Warten schrecken sie leider Anwender ab.

17.3 Nomachine NX

X-Programme wissen bei ihrer Ausführung auf einem X-Client nicht, wo sich der zugehörige X-Server befindet. Sie können bei einem über ein WAN erreichbaren X-Server nicht selbstständig *round trips* einsparen. Dies ermöglicht für sie die Erweiterung NX des X.11-Protokolls der Division Nomachine von Medialogic S.p.A.

NX beschleunigt den Aufbau von X-Bildschirmen durch

- Cachen (Speichern und aus dem Speicher wieder verwenden) bereits einmal übertragener Daten und differentielle Übertragung;
- Einsparen der meisten **X-round trips** und
- Verdichten der verbleibenden X-Kommunikation.

Durch seine enorme Verbesserung der Effizienz können Nutzer mit Network-Computing erheblich zufriedener sein als bei langsameren Technologien wie reinem X.11. Nomachine NX unterstützt Linux-, Solaris- und Windows-Anwendungs-Server sowie Windows-, Solaris-, Linux- und MacOS/X-Clients.

17.3.1 Das Prinzip von Nomachine NX

Nomachine setzt auf dem in der Unix-Welt bewährten Protokoll X.11 auf. Es schaltet je einen Stellvertreter (*Proxy*) zwischen das X-Programm auf dem X-Client und den X-Server auf dem Benutzer-Endgerät. Es spart in mehreren Schritten überflüssigen X-Verkehr:

- Mit dem X-Programm auf dem Linux-Anwendungs-Server (X-Client) spricht der NX-Proxy X.11 und täuscht ihm so vor, er sei der zuständige X-Server. Das Programm *nxagent* übersetzt auf dem NX-Proxy des Anwendungs-Servers als bildschirmloser Schatten-X-Server das X-Protokoll in das NX-Protokoll.
- Der NX-Proxy cacht alle Nachrichten und überträgt nur Unterschiede, eliminiert unnötige Frage-Antwort-Spiele und komprimiert den restlichen Datenverkehr. Der NX-Proxy auf dem Linux-Anwendungs-Server leitet dann alle empfangenen Daten des X-Programms per NX-Protokoll an den NX-Proxy auf dem Endbenutzer-PC weiter. Auf dem Verbindungsabschnitt vom NX-Proxy auf dem Linux-Terminal-Server zum NX-Proxy auf dem Benutzer-Endgerät benötigt NX durch die drei Sparmaßnahmen sehr wenig Bandbreite.

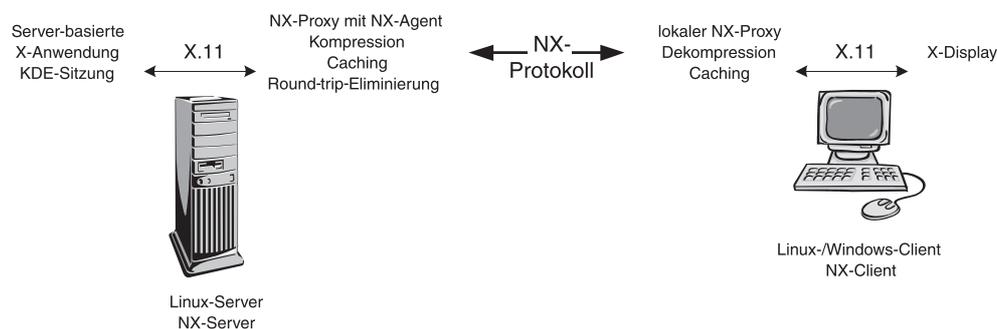


Abbildung 17.1: Komponenten von NX am Anwendungs-Server und am Benutzerarbeitsplatz

- Der NX-Proxy auf dem Benutzer-Endgerät benimmt sich gegenüber dem X-Server auf diesem wie eine ganz normale X.11-Anwendung: Der NX-Proxy und der X-Server sprechen wieder das ganz normale X-Protokoll über schnelle UNIX Domain Sockets. Der X-Server stellt Benutzern die Ausgaben des X-Programms auf dem Bildschirm des Client-PCs dar und nimmt Eingaben entgegen.

17.3.2 NX Nomachine-Test-Server aufsetzen

NX-Server lassen sich recht einfach einrichten. Die benötigten Programmpakete können Sie von <http://www.nomachine.com/download.php> beziehen. Der Hersteller bietet drei unterschiedliche Versionen an, die sich nur in der Anzahl der möglichen Verbindungen unterscheiden:

NX-Server Personal Edition: Diese Version ist auf einen Benutzer beschränkt und bietet sich daher zum gelegentlichen Administrieren von Linux-Servern von Windows-PCs an.

NX-Server Small Business Edition: Diese Version ist auf fünf Benutzer beschränkt. Sie wendet sich an Gruppen von Administratoren, die regelmäßig auf Anwendungs-Server zugreifen möchten und an kleinere Organisationen mit wenigen Außendienstmitarbeitern und Teleworkern, die gelegentlich auf Anwendungs-Server zugreifen wollen.

NX-Server Enterprise Edition: Diese Version ist praktisch nur durch die Ressourcen des Servers selbst beschränkt. Da sie sehr viele Anwender bedienen kann, empfiehlt sie sich für Server-basierte Datenverarbeitung über das Internet und als besserer Ersatz für Web-Services und Server-basierte Java-Applikationen.

Teile der Software von Nomachine fallen unter Open-Source-Lizenzen. Darüber hinaus entwickeln freie Projekte freie Versionen dieser Lösung. Auf seiner Web-Seite weist Nomachine auf diese Projekte hin.

Konfiguration und Funktionsumfang aller kommerziellen Versionen ist praktisch identisch. Nomachine unterscheidet noch Versionen für verschiedene Linux-Plattformen. Das folgende Beispiel verwendet die NX-Server Enterprise Edition für SuSE Linux Professional 9.1 auf einen SuSE Linux Professional 9.2.

Da die NX-Server-Komponenten auf Libraries der NX-Client-Komponenten aufsetzt, kann man sie erst auf Servern einrichten, wenn man dort bereits die NX-Client-Software installiert hat.

Beide Komponenten installiert man aus Quellen im RPM-Format.

Zwar deklariert Nomachine seine Software als mit SuSE 9.1 lauffähig. In den kurzen Tests der Redaktion gab es keine Probleme bei einer Installation auf SuSE Linux 9.2.

```

boss:~ #
boss:~ # rpm -Uvh nxclient-1.4.0-91.i386.rpm
Preparing... ##### [100%]
  1:nxclient ##### [100%]
boss:~ # rpm -Uvh nxserver-1.4.0-103.i386.rpm
Preparing... ##### [100%]
  1:nxserver ##### [100%]
NX> 700 Autodetected system 'suse'

...

Getting the evaluation license

Please visit the http://www.nomachine.com/evaluation/ site to
subscribe for an evaluation trial and get a valid license key.

Getting support

Please contact the NoMachine support at
http://www.nomachine.com/support/
or write an e-mail to support@nomachine.com by specifying the
customer id provided at the time of purchase.

NX> 700 Bye
boss:~ #

```

Zum Aktivieren der Software müssen Sie von <http://www.nomachine.com/download.php> einen 30-Tage-Lizenzschlüssel (*key*) anfordern. Den Schlüssel sendet Ihnen Nomachine an die von Ihnen zuvor angegebene Mail-Adresse. Kopieren Sie ihn in das Verzeichnis `/usr/NX/etc`, und belegen Sie die Datei mit den richtigen Rechten.

```

boss:~ #
boss:~ # gunzip key.tar.gz
boss:~ # tar -xvf key.tar
key.txt
boss:~ # mv key.txt /usr/NX/etc/
boss:~ # chmod 644 /usr/NX/etc/key.txt
boss:~ #

```

Standardmäßig installiert sich NX in das Verzeichnis `/usr/NX`. Die Software verwaltet man von einem Kommandozeilen-Interface (*Command Line Interface, CLI*) aus im Verzeichnis `/usr/NX/bin`. Nomachine hat ein grafisches Administrations-Interfaces angekündigt.

Prüfen Sie die erfolgreiche Lizenzeingabe via `nxserver -version`. Wenn NX die Lizenz richtig erkannt hat, zeigt es Version des NX-Servers an.

```
boss:~ #
boss:~ # /usr/NX/bin/nxserver --version
NXSERVER - Version 1.4.0-103 EE
boss:~ #
```

Da man die Programme vom Client aus konfiguriert, beschränkt sich die Administration des NX-Servers primär auf die User-Verwaltung. Sollen Benutzer auf den NX-Server zugreifen dürfen, benötigen sie zuvor einen lokalen Linux- oder PAM-Account. Bitte legen Sie zu Testzwecken wie im Kapitel 3 beschrieben z. B. mit YaST einen Benutzer *testuser* an.

Damit Betriebssystembenutzer via NX arbeiten können, müssen die lokalen Benutzer zusätzlich mit dem Befehl `/usr/NX/bin/nxserver --useradd USERNAME` in die NX-Anwenderdatenbank eingetragen werden. Nur mit gültigem Betriebssystem- und NX-Account können sich Benutzer via NX an einem NX-Server anmelden.

```
boss:~ #
boss:/usr/NX/bin # nxserver --useradd testuser
NX> 100 NXSERVER - Version 1.4.0-103 EE (Evaluation)
NX> 900 Setting password for user: testuser
NX> 102 Password:
NX> 102 Confirm password:
NX> 110 Password for user: testuser added to the NX password DB
NX> 802 The NX user DB is disabled. Any authenticated user is
  ↓ enabled to login
NX> 900 Adding public key for user: testuser to the authorized
  ↓ keys file
NX> 716 Public key added to: /home/testuser/.ssh/authorized_
  ↓ keys2
NX> 900 Verifying public key authentication for NX user:
  ↓ testuser
NX> 900 Public key authentication succeeded
NX> 999 Bye
boss:/usr/NX/bin #
```

Benutzer können ihre eigenen Passwörter via `/usr/NX/bin/nxserver -passwd` ändern.

Die zur Administration der Benutzer möglichen Befehle (Anlegen, Löschen, Aktivieren, Deaktivieren) des NX-Servers erhalten Sie als Ausgabe des Befehls

```
/usr/NX/bin/nxserver -usage.
```

17.3.3 NX-Client installieren und konfigurieren

Wie zuvor beschrieben richtet man Applikation am NX-Client ein und stellt dort auch die Komprimierung sowie den Bildschirmmodus ein.

Zum Zugriff auf den NX-Server gibt es mehrere Clients. Diese finden Sie unter <http://www.nomachine.com/download.php>.

Auf Linux- bzw. Unix-Endgeräten installiert man NX-Clients wie zuvor beschrieben via rpm (siehe Abschnitt 17.3.2). Für Windows-Endgeräte führen Sie zur Installation die Datei `nxclient-1.4.0-92.exe` aus.

Sowohl bei der Installation für X-Systeme als auch im Windows-Desktop finden Sie nun einen Programmpunkt *NX-Client für Windows bzw. Linux* im Startmenü der Desktops.

Außer verschiedenen Hilfsfunktionen (URL zur Internet-Hilfe von Nomachine) und einem Menüpunkt zur Deinstallation finden Sie drei verschiedene Programme:

- *NX-Client für Windows/Linux*: Hierbei handelt es sich um den eigentlichen NX-Client. Sie können Profile zu unterschiedlichen Anwendungs-Servern anlegen und diese mit einer Aufklapp-Box auswählen und sich durch Eingabe Ihres User-Namens und Ihres Passworts verbinden.
- *NX Connection Wizard*: Der *Connection Wizard* legt Profile an, die NX Clients verwenden können.
- *NX Session Administrator*: Der *Session-Administrator* dient zum Monitoring der lokalen Session. Sie können sich jederzeit einen Überblick über Ihre Sessions verschaffen und bei Bedarf Verbindungen beenden.

Wenn Sie den NX-Client zum ersten Mal starten, gelangen Sie automatisch zu dem *NX Connection Wizard*, der Sie beim Einrichten einer Applikation unterstützt.

Applikationen legt man in zwei Schritten an:

- Zunächst definieren Sie den NX-Server und die Verbindungsgeschwindigkeit zu Ihrem Server. Starten Sie hierzu den *Connection Wizard*. Nach einer Willkommensnachricht müssen Sie in einem ersten Schritt den Namen Ihres Profiles, den Verbindungs-/NX-Server sowie die Verbindungsgeschwindigkeit und den Verbindungs-Port definieren.
- In einem zweiten Schritt definieren Sie die Verbindungsart, die Applikation und die Darstellung.



Abbildung 17.2: Angaben zum NX-Host und der Verbindungsmethode

NX unterstützt zu diesem Zeitpunkt drei verschiedene Verbindungsarten: X.11 (*Unix/Linux*), RDP (*MS Terminal-Services*) und RFB (*VNC*). Wählen Sie hier die gewünschte Applikation, und spezifizieren Sie diese (z. B. KDE, Gnome, CDE).

Weiterhin können Sie in diesem Schritt die Darstellung der Applikationen festlegen. NX kann Anwendungssitzungen auf Benutzer-Arbeitsplätzen im Vollbild oder im Fenster darstellen:

Im Vollbildmodus bedeckt eine X-Sitzung den ganzen Bildschirm des Benutzer-PCs, und im Fenstermodus können Benutzer die Größe des Fensters selbst festlegen und dort entweder einen kompletten Desktop des Anwendungs-Servers oder nur einzelne seiner Anwendungen darstellen.

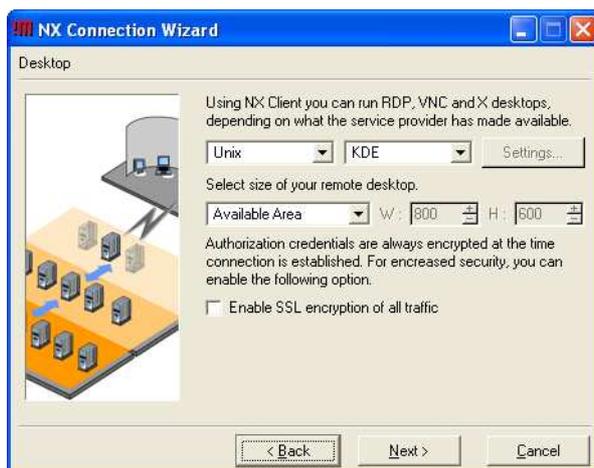


Abbildung 17.3: Angaben zur Applikation und Darstellung

Wenn Sie den Menüpunkt *Enable SSL Encryption of all Traffic* auswählen, verwendet NX ausschließlich Port 22 (*ssh*) zur Kommunikation. Weiterhin verschlüsselt SSH die komplette Kommunikation zum NX-Server. Diese Funktion wird meist verwendet, wenn sich die Applikations-Server hinter Firewalls befindet.

Nun können Sie die Verbindung Ihres Clients testen. Das Beenden des *Connection Wizards* startet direkt den NX-Client.

Mit Ihrem User-Namen und dem passenden Passwort können Sie sich nun an Ihrem Server anmelden und dort arbeiten.



Abbildung 17.4: Anmeldung am NX-Server

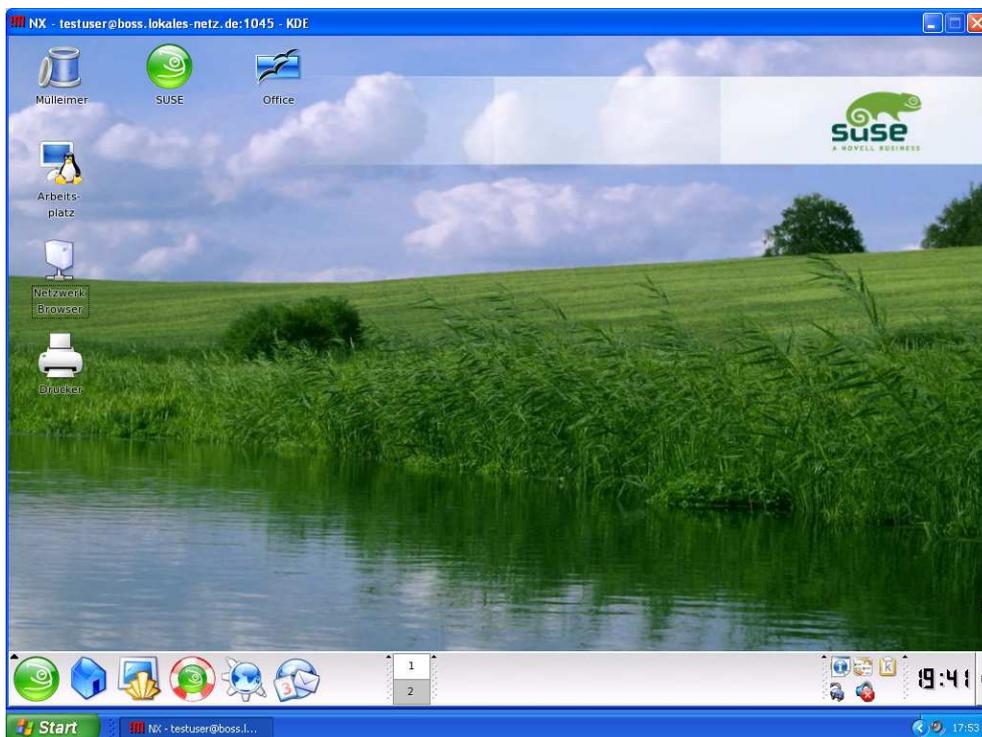


Abbildung 17.5: KDE via NX

NX erlaubt nicht nur, komplette Betriebssystemoberflächen abzubilden. Sie können auch einzelne Applikationen wie *xterm* oder *OpenOffice.org*-Applikationen anzeigen lassen.

Bitte verfolgen Sie das Anlegen eines separaten Profils. Starten Sie dazu erneut den *Connection Wizard*, und definieren Sie im ersten Schritt den Server *Boss* wie bereits im vorigem Beispiel, und benennen Sie dieses Profil *Office*.

Im zweiten Schritt wählen Sie nun bitte erneut *Unix/X11* als Protokoll und wählen jetzt nicht *KDE*, sondern den Menüpunkt *Custom*.



Abbildung 17.6: Definierte Applikationen mit NX

Definieren Sie nun die Anwendungen, die Sie ausführen möchten. Zum Darstellen einzelner Applikationen können Sie den Fenstermodus *Floating Windows* verwenden. Dieser ermöglicht Ihnen eine Bildschirmintegration wie bei lokal ablaufenden Fenstern.

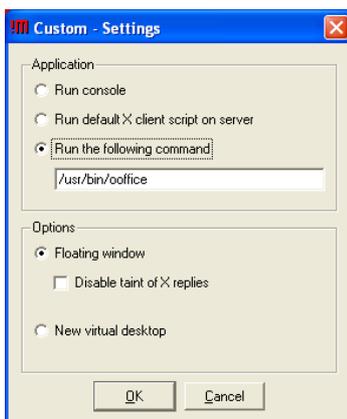


Abbildung 17.7: Definierte Applikationen *OpenOffice* mit NX.tif

So können Sie auch sehr einfach weitere Server/Applikationen ansprechen, auch wenn auf diesen Servern keine NX-Komponente abläuft. Hierzu müssen Sie nur das Display umlenken oder in einem Aufruf von SSH eine weitere Verbindung initialisieren.

Wiederholen Sie bitte den Schritt 1 erneut, und legen Sie ein neues Profil `xterm` auf dem Linux-Server `apps` an.

Tip: Bitte beachten Sie, dass das folgende Beispiel ein X.11-Forwarding via SSH voraussetzt. Die eingestellten Sicherheitsrichtlinien des Zielrechners `apps` verhindern unter Umständen eine Ausführung dieses Vorgehens.

Definieren Sie nun als ausführbares Programm die Applikation `SSH`, und rufen Sie die Applikation `xterm` auf dem Zielrechner als User `root` auf.

Dies erreichen Sie mit folgendem Befehl:

```
ssh root@apps.lokales-netz.de /usr/bin/xterm
```

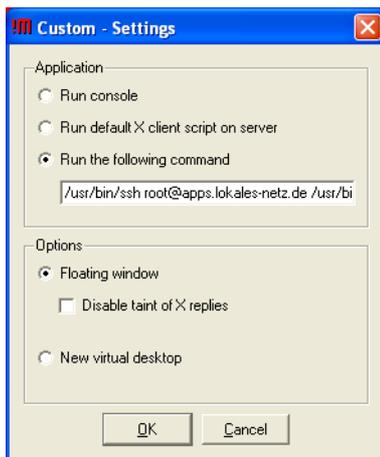


Abbildung 17.8: Definierte Applikation `xterm` auf `Apps`

Hier baut NX eine Verbindung zwischen den Rechnern `apps` und `boss` über SSH auf. X.11-Forwarding verweist nun den X.11-Strom auf den X-Client auf den NX-Server `boss` und sendet ihn und dann via NX an den NX-Client.

Wenn Sie sich für Sound, Drive- und Drucker-Mapping mit Nomachine NX interessieren, konsultieren Sie bitte die stetig wachsende Support-Datenbank von Nomachine.

17.4 Tarantella Secure Global Desktop Enterprise Edition

Tarantellas Middleware Secure Global Desktop Enterprise Edition verbindet nahezu beliebige Clients mit nahezu beliebigen Anwendungs-Servern. Statt Anwendungen lokal auf den Arbeitsplatzrechnern der Anwender zu installieren, stellt man sie diesen über ein LAN oder über ein WAN als Server-Dienst zur Verfügung, um Administrations-, Sicherheits- und Support-Vorteile zu erzielen und die Gesamtkosten der Datenverarbeitung zu senken. Tarantella nennt seine Middleware Secure Global Desktop Enterprise, kurz *SGD EE*, auch *Applikations-Portal*.

Tarantella SGD EE arbeitet als Middleware zwischen Anwendungs-Servern und Clients.

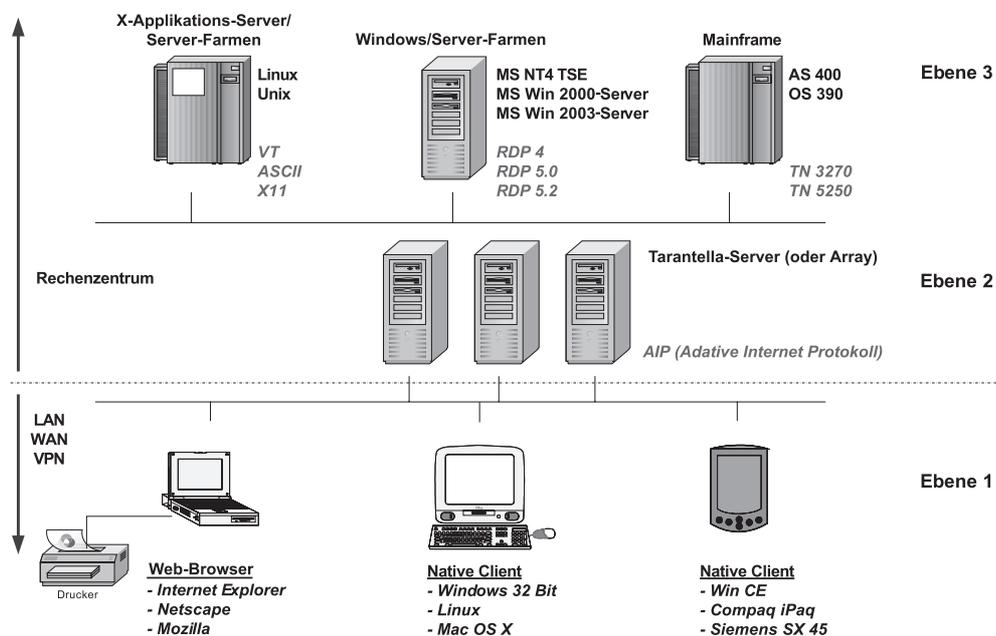


Abbildung 17.9: Die drei Ebenen aus Tarantella-Sicht

SGD EE-Server arbeiten in drei Schritten:

In der Client-Ebene, der Ebene 1, arbeiten die Anwender und in der Ebene 3 die Applikations-Server, typischerweise Windows-Terminal-Server, Unix/Linux- und Mainframe-Systeme, plattformunabhängige Java-Applikations-Server sowie File-, Proxy- und Print-Server (manchmal auch als Ebene 4 bezeichnet).

Zwischen der 1. und der 3. Ebene, also in der Ebene 2, fangen SGD EE-Server die Protokolle der 3. Ebene auf, übersetzen diese in ihr eigenes Protokoll, das Adaptive Internet Protokoll (AIP) und stellen dessen Inhalt bandbreitenoptimiert den Clients zur Verfügung.

Benutzersitzungen von Windows-Programmen im Terminal-Server-Modus kann es via RDP abfangen (siehe auch unten, Abschnitt 19.6). Windows-Terminal-Server kommunizieren mit Clients mit Microsofts proprietärem Remote Desktop Protokoll (*RDP*). Microsoft bietet jedoch selbst ausschließlich Windows-basierende Clients an. Aufgrund einer Lizenzvereinbarung zwischen Microsoft und Tarantella kennt Tarantella Microsofts Spezifikationen und Code zum Zugriff auf die Terminal-Server-Sitzungen. Der Unix/Linux-basierende SGD EE-Server verhält sich gegenüber den Terminal-Servern dann ebenfalls wie ein Microsoft Terminal-Server-Client.

Bei Unix/Linux-Applikations-Servern verhält sich der SGD EE-Server als X-Server, der den X.11-Datenstrom von einer X-Applikation auf einem Anwendungs-Server erhält.

SGD EE-Server können ihr eigenes Protokoll AIP (*Adaptive Internet Protokoll*) mit 128 Bit SSL optional verschlüsseln.

Auf Firewalls muss man für SGD EE keine zusätzlichen Ports öffnen, da man es für einen *Single Port Access* über einen Standard-Port wie Port 80 (*http*) oder Port 443 (*https*) konfigurieren kann. Auch Web-Proxy-Server mit Protokoll-Tunneling stellen keine Hindernisse für SGD EE dar. EE-Server entscheiden dynamisch nach der zum Client zur Verfügung stehenden Bandbreite, wie weit sie den Datenstrom verdichten.

AIP benötigt, falls erforderlich, weniger Bandbreite als Microsofts RDP: Während EE-Server in einem 10 MBit-LAN praktisch nichts komprimieren, verdichten sie den Datenstrom bei langsamen WAN-Verbindungen oder wenn sich mehrere Anwender eine Leitung teilen.

Verbindungen mit 19.200 bps (z. B. bei Handy-Verbindungen mittels GSM) reichen für einen Benutzer gerade noch aus, 64 KBit-ISDN-Leitungen bedienen bequem zwei User.

Die Benutzer der EE-Server kann man auf unterschiedlichste Art verwalten: mit Unix-Accounts auf den EE-Servern (z. B. NIS), Authentisierung gegenüber LDAP- oder Active Directory-Servern, Windows-Domänen, Web-Servern und/oder SecureID-Servern oder mit einer eigenen Benutzerverwaltung in SGD EE selbst.

Das Session-Resume unterstützt mobile Nutzer: Benutzer können nicht ordnungsgemäß beendete Verbindungen später wieder aufnehmen. Damit sind Tunnel wie auf der ICE-Strecke von Frankfurt nach Göttingen oder Staus im Elbtunnel für mobile GSM-Nutzer nicht mehr ganz so schrecklich.

Auch in mobilen und angeblich papierlosen Büros wollen Anwender drucken. EE-Server kann man als IP-Druck-Server ansprechen und so Druckaufträge ebenfalls über AIP verschlüsseln und komprimiert an EE-Clients schicken.

Da EE automatisiert Druckaufträge in PDF umwandeln und an den Client senden kann, kann man durch die hohe Komprimierungsrate von PDF-Dokumenten auch in WAN-Umgebungen noch vernünftig drucken.

Da EE nicht sehr Ressourcen-hungrig ist, kann man mit einem Single Pentium III, 1 GHz und 1 GB Hauptspeicher ohne Schwierigkeiten 70 User gleichzeitig anbinden.

Die Installation selbst nimmt etwa 180 MB Plattenplatz in Anspruch. Bei üppiger Hardware-Ausstattung sind bis zu 1.000 User pro Server möglich. EE nennt unter <http://www.tarantella.com/whitepapers/> die Testergebnisse verschiedener Betriebssysteme und Hardware-Lieferanten.

Als Client kann man praktisch jedes Gerät mit Browser verwenden. Für Clients reicht ein langsames Pentium-System ab Pentium II und 64 MB Arbeitsspeicher. Beim erstmaligem Verbinden zum EE-Server lädt der Browser 700 KB Java-Programme/C-Programme und installiert sie automatisiert, unabhängig der Client-seitigen Berechtigungen (bei Bedarf nur temporär im Arbeitsspeicher).

17.4.1 Zielgruppen für Tarantella SGD EE

Mit EE kann man Applikationen im Intranet und Internet für gelegentlichen oder häufigen Gebrauch bereitstellen:

- *Server-zentriertes-Computing*: Da viele Unternehmen und öffentliche Einrichtungen die ausufernden Kosten der Betreuung von Windows-Arbeitsplätzen nicht mehr tragen wollen, stellen sie vermehrt auf Server-zentrierte Datenverarbeitung um. Neue Anwendungen, Patches, Hot Fixes und Service-Packs müssen sie dann nur noch zentral im Rechenzentrum installieren, laufen lassen und pflegen. Sie verwenden SGD EE als Middleware zwischen Windows-Anwendungs-Servern und Arbeitsplatzgeräten, wenn diese über ein WAN zugreifen oder sie die Last der zahlreichen Arbeitsplatzgeräte auf mehrere Windows-Terminal-Server verteilen wollen.
- *PCX-Server oder RDP-Zugriff*: Immer mehr Unternehmen und öffentliche Einrichtungen verwenden aus Sicherheits- und Kostenüberlegungen möglichst Open-Source-Programme. Um bei Bedarf von Java-fähigen Browsern ihrer Linux-Clients auf Windows-Programme zugreifen zu können, ohne auf Clients virenanfällige und instabile Microsoft-Produkte zu installieren, stellen sie mit EE und Windows-Terminal-Servern die in der Übergangszeit noch gewünschten Windows-Programme bereit. EE ermöglicht es Anwendern, von den verschiedensten Clients aus auf den verschiedensten Applikations-Servern (z. B. *Gnome* auf Solaris) zu arbeiten. Andererseits ist es auch möglich, Unix/Linux-Applikationen via AIP auf Windows-Clients darzustellen. So können bestehende Windows-Clients weiterverwendet werden und Anwender bereits auf X-basierende Applikations-Server zugreifen.

- *Sicherheit*: Anwender greifen immer und ausschließlich über den Tarantella-Server auf Anwendungen und Daten zu. Dies ermöglicht es, ein sehr sicheres und sehr einfach zu kontrollierendes Netz aufzubauen. Es ist nicht erforderlich, zusätzliche Ports zum Zugriff auf Rechenzentren zu öffnen. Jede Kommunikation zwischen Client und EE-Server kann zusätzlich verschlüsselt sowie bei Bedarf aufgezeichnet werden.
- *Webenabler*: Viele Firmen entwickeln nur noch Applikationen, die in Browsern lauffähig sind. Die Bedeutung von Browsern als der universellen zentralen Anwendung wächst. Mit EE kann man verschiedenste Anwendungen innerhalb weniger Minuten Browser-fähig gestalten und so Usern z. B. X.11-Anwendungen über Portale einheitlich anbieten.

17.4.2 Secure Global Desktop Enterprise Edition installieren

EE selbst läuft derzeit ausschließlich auf einem Linux ab Kernel Version 2.6 oder Sparc Solaris ab Version 8. Als Standardinstallation empfiehlt sich eine »normale« Linux-Installation mit installiertem X-Grundsystem. Als Standardinstallation empfiehlt sich eine »normale« Linux-Installation mit installiertem X-Grundsystem, zusätzlich müssen Sie nur das Paket *compat-gdm* nachinstalliert haben.

Die Installation der EE richtet außer den benötigten EE-Komponenten einen durchkonfigurierten Web-Server und eine Tomcat-Erweiterung ein.

Anders als in früheren Versionen empfiehlt es sich, ausschließlich mit dem von Tarantella mitgelieferten Web-Server/Tomcat zu arbeiten. Bestehende Web-Server und/oder Tomcat-Versionen können jedoch parallel weiterbetrieben werden.

Bei der Installation von EE ist das Netzwerk selbst eine Hauptfehlerquelle. Prüfen Sie bitte das Netzwerk einschließlich Namensauflösung (*ping*, *nslookup/dig* etc.) vor der Installation von Tarantella, da nachträgliches Ändern des primären Host-Namens sehr zeitaufwändig ist.

Typischerweise sollte daher neben dem *localhost* der Full Qualified Domain Name (*FQDN*) und der Shortname des SGD EE-Servers in der */etc/hosts* aufgeführt sein:

```
127.0.0.1localhost
192.168.1.2boss.lokales-netz.de    boss
```

Tarantella bietet zwei kostenlose Möglichkeiten, seinen SGD EE nach dem Download von <http://www.tarantella.com/download> zu testen:

- eine Zwei-Benutzer-Version mit Registrierung, aber ohne zeitliche Begrenzung,

- eine 30 Tage lauffähige Demoversion für eine unbegrenzte User-Anzahl. Tarantella verlängert die Version auf Wunsch um weitere 30 Tage.

Um die Installation zu testen, loggen Sie sich als *root* in Ihr Linux-System ein.

Die SGD EE-Prozesse auf dem Tarantella-Server laufen unter dem Benutzer *ttaserv* als Mitglied der Gruppe *ttaserv*, die Web-Server- und die Application-Server-Komponenten unter *ttasys*.

Diese Benutzer sowie die Gruppe müssen Sie vor der Installation auf dem EE-Server wie folgt einrichten:

```
boss:~ # groupadd ttaserv
boss:~ # useradd -g ttaserv -s /bin/sh ttaserv
boss:~ # useradd -g ttaserv -s /bin/sh ttasys
```

Da EE die Benutzer ausschließlich für systeminterne Zwecke verwendet, müssen Sie keine Passwörter definieren. Die Accounts gelten dann als *Locked* – die Benutzer können nicht für einen externen Zugriff z. B. via *ssh* verwendet werden.

Ein Shell Script installiert SGD EE weitestgehend automatisiert. Es prüft, ob Sie das richtige Paket verwenden, ob genügend Festplattenspeicher zur Verfügung steht und ob die Benutzer und die benötigte Gruppe existieren. Fehlermeldungen während der Installation trägt das Skript in die Datei */tmp/ttainst.log* ein.

Rufen Sie zur Installation die heruntergeladene Datei mit *sh* auf, und verfolgen Sie diese.

```
boss:~ # sh ttai3li.shx
-----
Tarantella Secure Global Desktop Enterprise Edition
-----
Setup is verifying the shx file...
Setup is checking for disk space requirements...
Setup is installing package programs in /usr/package...
Secure Global Desktop for Intel Linux kernel 2.4+
Preparing for installation...
-----
Secure Global Desktop Software License Agreement
-----
To use Secure Global Desktop you must agree to be bound by
the terms of the Software License Agreement.

  Y - I have read, and accept the terms of the license agreement
  N - I do not accept the terms of the license agreement
  R - Let me read the license agreement

Accept terms of Software License Agreement? [Y]
```

Bestätigen Sie wie hier im Beispiel die Lizenzvereinbarungen, und geben Sie den Installationsort, den Archivierungszeitraum für Fehlermeldungen und den Lizenzschlüssel (Standard: Evaluierung) ein.

Wenn bereits ein Web-Server auf Port 80 läuft, fragt Tarantella, auf welchem Port sein eigener Web-Server laufen soll. Wenn Sie Ihren bereits installierten Web-Server nicht deaktivieren möchten, müssen Sie einen eigenen Zugriffs-Port für EE definieren, zum Beispiel Port 8080. Sollte EE feststellen, dass der Port nicht belegt ist, würde es diesen automatisch für seinen eigenen Web-Server reservieren.

```
-----
Setting up Tarantella Secure Global Desktop Enterprise Edition
-----
```

```
Secure Global Desktop Setup recommends you use the following
settings:
```

```
Installation type = install 4.00.901
Installation directory = /opt/tarantella
Peer DNS name = boss.lokales-netz.de
License mode = Evaluation (30-day limit)
HTTP port = 80 [not currently in use]
Archive logs every week? = yes (Sunday 03:00 hours)
```

```
Are these settings OK?
```

```
Y - Yes, install using these settings
N - No, tell me more about the options and let
me change the settings
Q - Quit now
```

```
OK to use these settings? [Y]
```

Auf manchen Systemen schlägt Ihnen das Installationsprogramm */usr/tarantella* als *Installation directory* vor, Sie können diese Vorgabe aber immer nach Ihren Vorstellungen anpassen.

Nach Bestätigung mit *y* beginnt die eigentliche Installation, die je nach Hardware etwa 15 Minuten dauert:

```
Secure Global Desktop Setup is recording settings...
```

```
...
```

```
Successfully installed Tarantella Secure Global Desktop
Enterprise
```

```
You have created a Secure Global Desktop CUPS printer.
Ensure you have enabled
CUPS raw printing by editing /etc/cups/mime.convs and
/etc/cups/mime.types.
```

```

These files contain comments explaining how to do this
(search for 'raw').

-----
What's next?
-----
Secure Global Desktop is now installed and ready to use.

To get started:
- In a web browser, go to:
  http://boss.lokales-netz.de/
- When prompted, log in with username "Administrator" and
  root's password.
- On your webtop, click Administration Guide to learn
  more about the product
  (we strongly recommend you read the "Getting started"
  section).
  Or click Object Manager to start creating user webtops.

To add license keys, click Array Manager on your webtop or type:
/opt/tarantella/bin/tarantella license add <key>...

For more information on Secure Global Desktop and SSH, see the
Administration Guide.

boss:~ #

```

Damit ist die EE-Installation abgeschlossen.

Wie im Abspann erwähnt muss für das korrekte Funktionieren der SSH-Dienst konfiguriert werden. Hierbei geht es im Wesentlichen um die Sicherheitseinstellungen, die standardmäßig eine Weiterleitung des X.11-Datenstroms verhindern.

Um diese zu ermöglichen, ändern Sie bitte in der Datei `/etc/ssh/ssh_config` die Zeile `ForwardX11 no` in `ForwardX11 yes` :

```

# Configuration data is parsed as follows:
# 1. command line options
# 2. user-specific file
# 3. system-wide file
# Any configuration value is only changed the first time it is
# set. Thus, host-specific definitions should be at the
# beginning of the configuration file, and defaults at the end.
# Site-wide defaults for various options

Host *
# ForwardAgent no

```

```
ForwardX11 yes
```

```
...
```

Weiterhin geht SGD EE von einer Installation auf Basis von IPv4 aus. Dies muss in der Datei `/etc/sysconfig/ssh` ebenfalls definiert werden.

```
## Path:          Network/Remote access/SSH
## Description:  SSH server settings
## Type:         string
## Default:      ""
## ServiceRestart: sshd
#
# Options for sshd
#
SSHD_OPTS="-4"
```

Beachten Sie bitte <http://www.tarantella.com/support/updates> für Tarantella-Updates und dort den Unterpunkt **Knowhow** für besondere Einstellungen und Tuning-Möglichkeiten abhängig vom verwendeten Betriebssystem.

Nach diesen Änderungen sollten Sie den SSH-Dienst via `rcsshd restart` und SGD EE via `/opt/tarantella/bin/tarantella restart --kill` neu starten.

```
boss:~ # rcsshd restart
Shutting down SSH daemon                               done
Starting SSH daemon                                   done
boss:~ # /opt/tarantella/bin/tarantella restart --kill
Trying to kill Secure Global Desktop services...
Starting Secure Global Desktop server (version 4.00.901).
Please wait...
Secure Global Desktop services are now available on this host.
boss:~ #
```

17.4.3 Erste Verbindung

Sie können sich nun mit einem Browser auf der Einstiegsmaske einen ersten Überblick über die Zugriffsmöglichkeiten auf SGD EE verschaffen. Die möglichen Funktionen sowie die Anwender-Performance können in Abhängigkeit des gewünschten Zugriffs variieren:

- **Log in (tcc-Client):** Der Menüpunkt verweist auf `http://<Servername>/sgd/index.jsp` (der direkt via `http://<Servername>/sgd` aufgerufen werden kann). Hierbei wird zunächst automatisch eine Java-Programm (*tcchelper*) vom SGD EE-Server geladen, der in einem weiteren Schritt eine EE-Komponente (*tcc, Tarantella Client Component*) vom Server lädt und installiert. Eine Anwendereingriff ist dafür nicht nötig. Sollten die Komponenten bereits installiert sein (durch vorherigen Besuch auf einem SGD EE-Server), so entfällt dieser

Schritt. Bei einem Verbindungsaufbau zu einem EE-Server wird automatisiert zunächst via `tcchelper` und einer Java Runtime-Umgebung die `tcc` ausgeführt. Die `tcc`-Komponente ist für den Datenverkehr des AIP-Protokolls zuständig. Diese Client-Variante ist seit Version 4 neu und bietet gleichzeitig den vollen Funktionsumfang bei ähnlicher Anwender-Performance wie der Native-Client. Die folgenden Darstellungen verwenden ausschließlich den `tcc`-Client.

- *Log in (classic webtop) (Java client)*: Dieser Menüpunkt verweist auf `http://<Servername>/tarantella/cgi-bin/secure/ttawlogin.cgi/?action=start&pg=index2.html` (der direkt via `http://<Servername>/tarantella` aufgerufen werden kann). Dieser Client ist komplett Java-basiert und läuft ebenfalls in einer Java Runtime-Umgebung ab. Tarantella führt diesen Client primär aus Kompatibilitätsgründen weiter. Vorteil dieser Lösung ist, dass die Kombination Browser/Java praktisch jede Betriebssystemvariante unterstützt.
- *Secure Global Desktop (Native Client)*: Im Gegensatz zu dem vorher genannten Client benötigt dieser Client kein Java Runtime-Umgebung und daher auch nicht so viele Client-seitige Ressourcen. Dieser Client wird daher oft auf Thin Clients verwendet, steht jedoch nur für die Betriebssysteme win32 Bit, Linux und Solaris zur Verfügung.



Abbildung 17.10: SGD EE-Einstiegsseite

Nachdem Sie sich für eine Client-Variante entschieden haben, können Sie sich nun via root-Account auf Ihrem EE-Server anmelden.



Abbildung 17.11: Anmeldeschirm TCC-Client

Nach dem Anmelden sehen Sie im Browser Ihres Clients zunächst den Standardbildschirm von EE mit einigen während der Installation angelegten Anwendungen (*xclock*, *Konsole*, *KDE* oder *Gnome* usw.) sowie Hilfe-Tools für Administratoren und Anwender (wie *Configuration Wizard*, *Object Manager*, *Arraymanager* usw.).

Links im Bild sehen Sie das so genannte *Launchpad*. Anwender können über diese Leiste alle Anwendungen durch Klicken starten. Im freien Arbeitsbereich sehen Anwender generelle Informationen oder die von ihnen aufgerufene Anwendung.

Hinweis: Da die Client-Darstellung vollständig durch HTML gesteuert ist, kann man diese nach Belieben für den Browser-Zugriff anpassen. Die folgenden Beispiele gehen immer von den »Standardansichten« aus.

Für den ersten Test mit Anwendungen des Applikations-Servers per Browser reichen die bei der EE-Installation automatisch eingerichteten Anwendungen. Anwender können diese als Vorlage für weitere Applikationen nutzen.

Rufen Sie die gewünschten Anwendungen über deren Icon auf. Der Benutzer-Client baut dann automatisiert über den EE-Server eine Verbindung zu dem Applikations-Server (hier dem EE-Server selbst) auf und durchläuft dabei die Anmeldeprozedur am Applikations-Server. Standardmäßig versucht Tarantella, die Zugangsdaten am GSD EE-Server auf die Applikation anzuwenden. Wenn der Applikations-Server

diese Zugangsdaten akzeptiert, fragt der EE-Server nicht erneut nach einem Login-Namen/Passwort für den Start von Applikationen.

Tip: Sie können beliebig viele Applikationen gleichzeitig ablaufen lassen. Unterhalb des Applikations-Icons können Sie Applikationen pausieren (in den Hintergrund schieben), abspielen (in den Vordergrund holen) oder beenden. Die Funktionsknöpfe sind hierbei denen eines Kassetten-/Videorecorders nachgebildet.

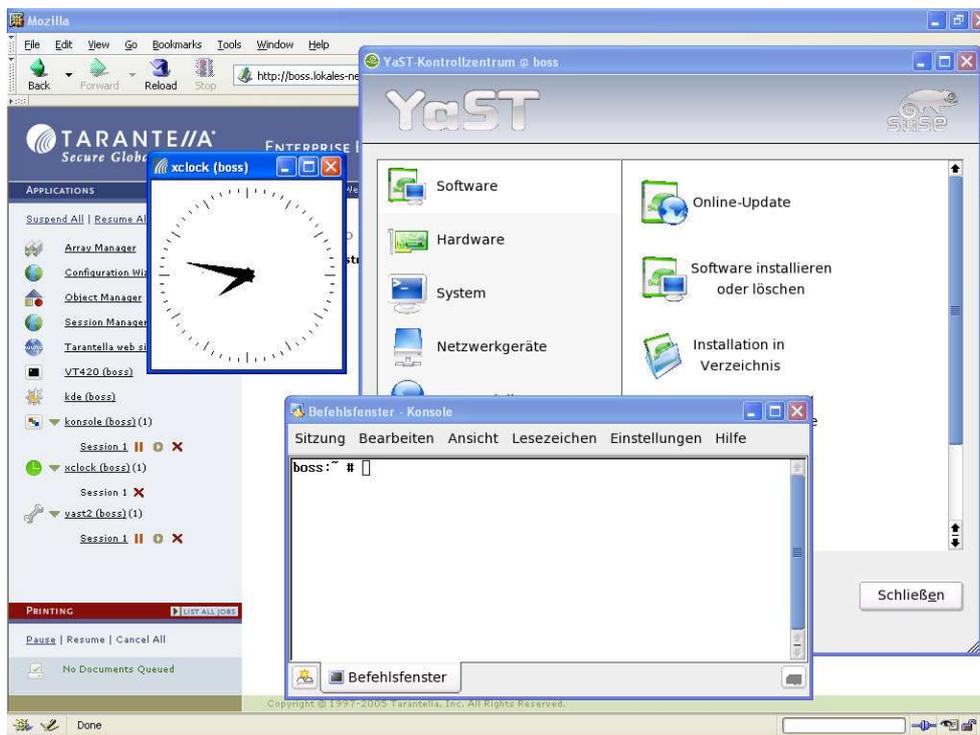


Abbildung 17.12: Tarantella mit verschiedenen Applikationen

Beenden kann man Anwendungen wie bei Linux üblich (z. B. über *Datei* • *Beenden* oder über die Eingabe von `exit` im Terminal-Fenster). Sobald Benutzer sich aus EE ausloggen, beendet EE automatisch deren Anwendungen, falls Sie EE nicht anders konfigurieren.

EE unterscheidet die Darstellung im *Webtop* und die Darstellung als *Independent Windows/Kiosk-Modus bzw. Vollbilddarstellung*. Sie können den Fenstermodus bei Browser-Zugriff jederzeit wechseln, indem Sie bei gedrückter `[Strg]`-Taste nochmals auf das Icon der Anwendung im *Launchpad* klicken.

Wenn Sie sich mit bestimmten Login-Daten an einer Applikation anmelden möchten, müssen Sie den durch EE angelegten lokalen Passwort-Cache umgehen. Um den Passwort-Cache während des Applikations-Starts nicht abzufragen, können Sie während des Klickens auf das Anwendungs-Icon (also beim Programmaufruf) die -Taste drücken. Dann fragt der EE-Server Sie nach einem User-Namen und einem Kennwort für das System. EE wendet dann die EE-Anmeldedaten oder gespeicherten Kennwörter nicht an.

Anwender können ihre Anmeldedaten speichern, um diese beim nächsten Aufrufen dieser Anwendung nicht erneut eingeben zu müssen. Die EE speichert das Passwort verschlüsselt.

Um sich vom EE-Server abzumelden, können Sie das Kreuz am rechten oberen Rand des Webtops betätigen (*Logout*). Nach einer Bestätigung gelangen Sie in das Standardabmeldefenster des EE. Je nach Konfiguration beendet EE nun die Anwendungen oder versetzt sie in einen Schlafzustand.

17.4.4 Tarantella konfigurieren und administrieren

EE hat beim Installieren abhängig von Ihren Eingaben ein Installationsverzeichnis angelegt. Dieses gliedert sich in vier Unterverzeichnisse (*/bin*, */etc*/, */var*, */web-server*):

/bin: Sie können EE über die Command-line (*CLI*) oder über grafische Administrations-Tools konfigurieren.

Um EE über die Kommandozeile zu steuern, wechseln Sie in das Installationsverzeichnis und geben dort den Befehl `tarantella` gefolgt von einem weiteren Parameter ein. Gerade beim Anlegen mehrerer Objekte kann man mit ein wenig Übung sehr viel schneller zum Ziel kommen als über das grafische Hilfe-Tool. Verschaffen Sie sich durch die Eingabe von `tarantella help` einen Überblick:

```
boss:~ # cd /opt/tarantella/bin/
boss:/opt/tarantella/bin # ./tarantella help
Usage: tarantella <command> [<command-specific args>]
```

```
Available commands:
  archive           Archives the server's log files
  array             Creates and manages arrays of Secure
                   ↙ Global Desktop servers
  arraymanager      Starts Array Manager
  config            Edits array-wide and server-specific
                   ↙ configuration
  emulatorsession  Lists and controls emulator sessions
  help             Displays this list of commands
```

```

license      Adds, lists and removes Secure Global
             ↵ Desktop license keys
object       Manipulates objects in the datastore
objectmanager Starts Object Manager
passcache   Manipulates the password cache
print       Controls Secure Global Desktop printing
             ↵ services
query       Examines the server's log files
restart     Restarts Secure Global Desktop services
role        Configures role occupants and their extra
             ↵ webtop links
setup       Changes Setup options, restores original
             ↵ objects
start       Starts Secure Global Desktop services
status      Shows the current status of Secure Global
             ↵ Desktop array members
stop        Stops Secure Global Desktop services
tscal       Lists, frees and returns Terminal
             ↵ Services CALs
uninstall   Uninstalls Secure Global Desktop from
             ↵ this host
version     Displays versions of installed Secure
             ↵ Global Desktop packages
webserver   Controls the Secure Global Desktop
             ↵ Web Server
webtopsession Lists and controls webtop sessions

Use "tarantella <command> --help" to get help on a command.

boss:/opt/tarantella/bin #

```

/etc: In diesem Verzeichnis finden Sie Konfigurationsdateien für EE.

/var: In diesem Verzeichnis befinden sich die User-Daten, Applikationskonfigurationen, Zuordnungen sowie alle Änderungen am Webtop, Startfenster usw.

/webserver: In diesem Verzeichnis speichert EE alle Dateien zur Tarantella-Version des Apache Web-Servers sowie der Tomcat-Erweiterung.

Um das Risiko zu begrenzen, dass Ihre Konfigurationsmühen vergebens sind, sollten Sie zumindest das Verzeichnisses /var regelmäßig sichern, besser das ganze Verzeichnis /opt/tarantella.

SGD bietet Administratoren (*root*) im angemeldeten Tarantella-Fenster zusätzliche Applikations-Icons wie das Tarantella-Hilfe-Dokument *Administration-Guide* sowie die Tarantella-Konfigurations-Tools *Array Manager* und *Object Manager* und auch mehrere *Wizards*. Letztere vereinfachen das Einstellen der Parameter für Applikationen und User erheblich.

Mit dem *Array Manager* konfiguriert man einmalig allgemeine Einstellungen wie z. B. Lizenzschlüssel, Farmen (*Arrays*) von Tarantella-Servern, Port-Definitionen, Anmeldeverhalten usw.

17.4.5 User für Tarantella anlegen

Die Benutzerverwaltung ist sehr vielschichtig, da sie auch den Zugriff sehr vieler tausend User unterstützen kann. Tarantella bietet acht verschiedene Arten einer User-Authentifizierung. Am üblichsten ist die Authentifizierung gegenüber lokal angelegten Unix-Accounts, Windows NT-Domänen, Web-Servern, SecureID oder gegenüber einem Verzeichnisdienst wie LDAP oder dem Microsoft ADS-Server.

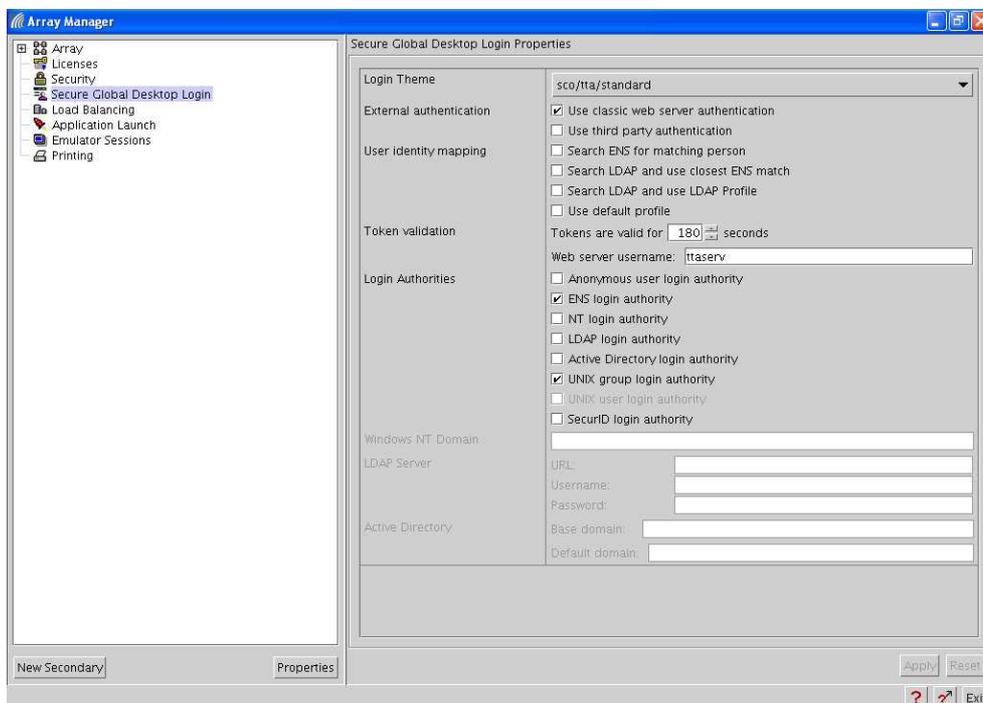


Abbildung 17.13: User-Verwaltung in SDG EE

Ein dedizierter Server zur Benutzerverwaltung empfiehlt sich vor allem bei hohen User-Zahlen oder wenn User auf mehrere unterschiedliche Applikations-Server zugreifen müssen. Damit legt man die Daten der Benutzer nicht auf allen Applikations-Servern an, sondern man definiert einmalig auf einem Verzeichnis-Server User-Profile, Mail-Adresse, Dateiablage, Rechte usw.

Voreingestellt ist eine Authentifizierung gegenüber einem Unix-Account. Alle User mit einem Unix-Account auf dem Tarantella-Server können sich an Tarantella an-

melden. Möchten Sie (für Tarantella) weitere User anlegen, so verwenden Sie am besten die Benutzerverwaltung von YaST. Das Linux-Kennwort ist gleichzeitig das Kennwort für Tarantella. Um damit Erfahrungen zu sammeln, legen Sie bitte in YaST einen User *testuser* an, und melden Sie sich mit dessen Namen an Tarantella an (wie bereits erläutert).

17.4.6 Applikationen zuordnen

Der Unterschied zwischen der Oberfläche des zuvor verwendeten Users *root* und dem soeben eingerichteten *testuser* ist leicht zu erkennen. Dem *testuser* fehlen die administrativen Tools von EE. Nicht-*root*-Anwender dürfen weder auf den Object/Array-Manager noch auf die *Wizards* zugreifen.

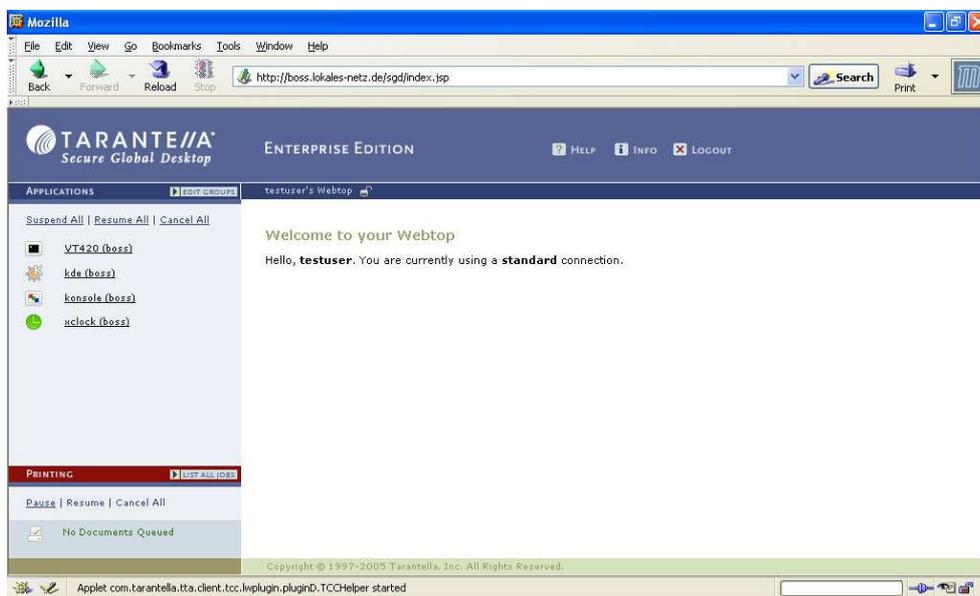


Abbildung 17.14: Anwender-Desktop

Administratoren legen im *Object-Manager* fest, welche Applikationen bestimmte User/Usergruppen sehen dürfen.

Die Installation von EE legt dazu drei Grundstrukturen an. Diese sind im *Object-Manager* leicht als Wurzelverzeichnisse zu erkennen (siehe Abbildung *Profiles und Anwenderzuordnung*):

- *Organization*: User- und Applikationen-Objekte. Administratoren können zum Gliedern eigene Organisationseinheiten (*OUs*) bestimmen.

- *Tarantella System Objects*: Hier legt die EE Systeminformationen wie den Array-Manager und den Object-Manager ab und verwaltet Administratorrechte. Diesen Zweig kann man nur bedingt ändern.
- *Dc=de*: Bei Verwendung von Verzeichnisdiensten können die Strukturen, z. B. Object Units, Container, Gruppen usw. in EE nachgebildet werden. Diese Gruppen kann man dann separiert für EE konfigurieren und in diesem Zweig anlegen.

Hinweis: Sowohl der Object- als auch der Array-Manager laufen auf dem Server in einem Tarantella-eigenen Java Runtime Environment. Aus diesem Grund gestaltet sich die Arbeit mit diesen stellenweise etwas träge. Achten Sie daher bei der Arbeit mit dem grafischen Administration-Tool darauf, alle Änderungen wie eingestellt durch Mausklick auf *Apply* zu übernehmen.

Öffnen Sie im *Object Manager* den Verzeichniszweig *Tarantella System Objects*. Dort hat jedes Anmeldeverfahren für EE ein eigenes, voreingestelltes Benutzerprofil. Wenn ein Benutzer sich in Tarantella anmeldet, entscheidet der Tarantella-Server selbstständig abhängig von der gewählten Authentifizierung (Unixgruppen, NT-Domänen, LDAP-Profil), welches Profile und somit welche Anwendungen er dem Tarantella-User zuweist.

Wählen Sie das Profile *Unix User Profile* durch Doppelklick aus. Die anzuzeigenden Anwendungen definieren Sie im rechten Fenster auf der Registerkarte *Links*.

Um eine eigene Applikationen zuordnen zu können und die durch Tarantella definierten Standardapplikationen auszublenden, markieren Sie mit der rechten Maustaste die *OU o=organization* im rechten Fenster, und wählen Sie *Remove*. Das Anzeigefenster der Registerkarte *Links* ist nun leer. Unix-Anwendern, die sich nun bei Tarantella anmelden würden, stünden keine Applikationen zum Starten zur Verfügung.

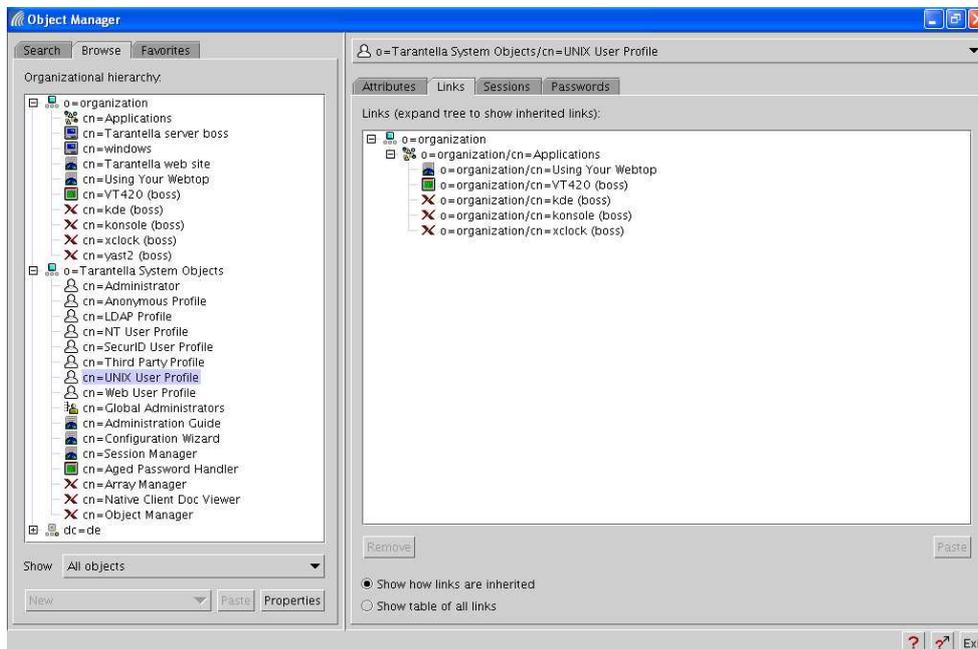


Abbildung 17.15: Profiles und Anwendungszuordnung

Öffnen Sie nun gleichzeitig den Verzeichniszweig `o=organization`. Sie sehen die bei der Installation von Tarantella angelegten Applikationen (z. B. `xclock`, `VT420` usw.). Sie können diese Applikationen nun mit der Maus in die Registerkarte *Links* ziehen und somit die Anwendungen den Unix-Usern unter EE zur Verfügung stellen. Sie können über diesen Weg pro Benutzer bis zu 1.024 Anwendungen bzw. Profile zuordnen.

17.4.7 Unix-Applikationen definieren

Sie können Applikationen über mehrere Wege anlegen. Das CLI-Interface wird bevorzugt zur Skript-gesteuerten Konfiguration verwendet und der OM, um Applikationen im Detail zu konfigurieren. Für neue Applikationen verwenden Sie am besten den *Configuration Wizard*.

Eine neue Applikation legt man in drei Schritten an:

1. Definieren Sie den Host (Applikations-Server), auf dem die Anwendung ablaufen soll (oder verwenden Sie den bereits angelegten).
2. Definieren Sie die Anwendung, und verknüpfen Sie diese mit dem unter 1. definierten Host.
3. Weisen Sie die neue Applikation den Usern/User-Gruppen zu.

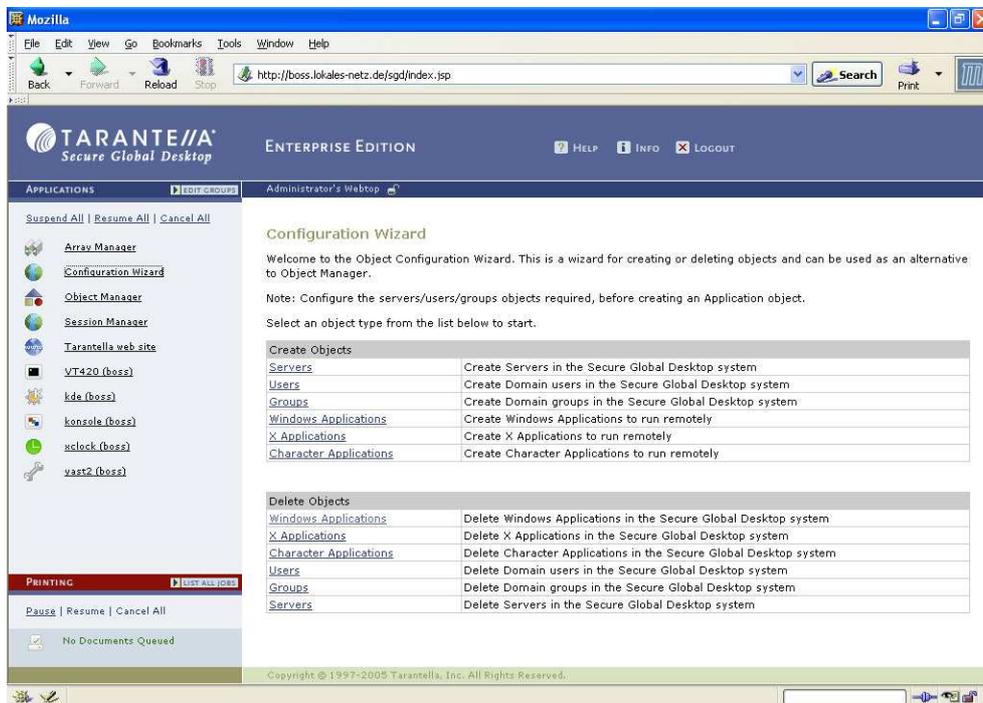


Abbildung 17.16: Configuration Wizard

Beliebige neue Objekte definiert man in Tarantella immer auf demselben Weg, der durch den *Wizard* vorgegeben wird.

Erstellen Sie ein neues Server-Objekt, indem Sie den entsprechenden Punkt im *Configuration Wizard* auswählen und den Anweisung des *Wizards* folgen. Felder mit »*« sind Pflichtfelder.

Dieser Server wird uns im nächsten Konfigurationspunkt als Applikations-Server angeboten.

- *Name*: Geben Sie dem neu erstellten Objekt einen beliebigen Namen.
- *Description*: Aussagekräftige Kurzbeschreibung.
- *Adress*: Tragen Sie hier den vollständigen Host-Namen ein. Testen Sie die Erreichbarkeit dieses Servers zuvor am besten mit `ping` und `nslookup`.
- *Windows NT Domain*: Bei Verwendung von Windows-Applikations-Servern tragen Sie die Domäne oder den Rechnernamen ein. Dies entfällt bei Unix-/Linux-Servern.
- *Location*: Wenn Sie mit sehr vielen Servern arbeiten, können Sie diese via *Location* gruppieren.

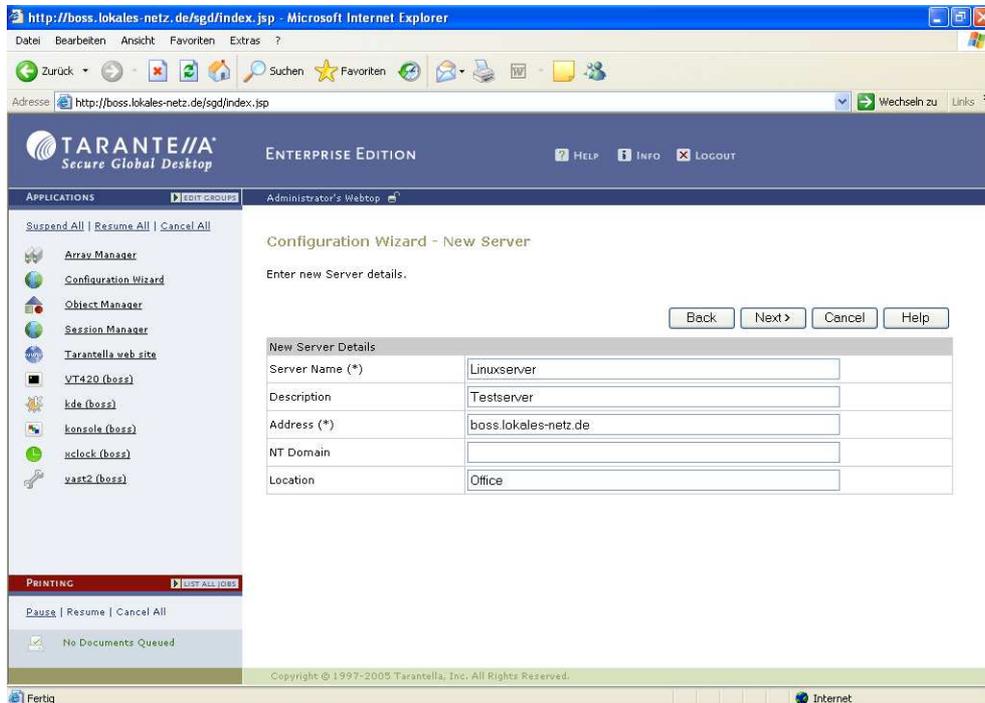


Abbildung 17.17: Neuer Host

Als zweiten Schritt zum Definieren einer Anwendung müssen Sie diese selbst auf demselben Weg wie zuvor den Server definieren. Rufen Sie daher im *Configuration Wizard* den Punkt *X-Application* auf, und geben Sie erneut die durch »*« gekennzeichneten Pflichtangaben ein.

In Abhängigkeit des ausgewählten Applikation wird Ihnen ein spezielles Formularfeld angezeigt.

Application Information: (Allgemeine Applikationsbezeichnungen):

- *Name:* Geben Sie dem neu erstellten Objekt einen aussagekräftigen Namen.
- *Description:* Aussagekräftige Kurzbeschreibung
- *Application:* Tragen Sie hier die Anwendung inkl. Pfad ein. Beispiele: /opt/kde2/bin/startkde (für eine KDE-Session), /usr/dt/bin/dtsession (CDE-Session), /sbin/soffice (StarOffice), /sbin/xeyes (Spiel) usw.

Application Defaults (Standardgröße)

- *Width & Height:* Es gibt verschiedene Fenstermodi zum Darstellen von Applikationen. Wenn Sie mit festen Größen arbeiten möchten, können Sie diese hier definieren.

Application Resumability (Fortsetzung von Anwendungen)

- *Resumable:* Hier legen Sie fest, was mit den Prozessen der Applikation geschehen soll, wenn Benutzer diese nicht ordnungsgemäß beenden (z. B. bei Verbindungsverlust einer GSM-Verbindung).
- *Never:* Tarantella beendet die Verbindung bei Verlassen des Screens.
- *Tarantella Webtop Session:* Die Verbindung bleibt so lange erhalten, wie man in Tarantella angemeldet ist.
- *Always:* Die Prozesse der Applikation bleiben so lange erhalten, bis Benutzer sie ordnungsgemäß beenden (z. B. über *Datei • Beenden*).

Application Display (Fenstermodus)

- *Webtop:* Tarantella zeigt das Programm im freien Bereich neben dem *Launchpad* an. Die Größe definieren Sie in den nächsten Zeilen unter *Width* und *Height*.
Dieser Modus eignet sich zur Ansicht kleiner Darstellungen wie *xclock*, *calc*, *YaST*, *VT420* und *Mainframe-Masken*.
- *Client Windows Management:* X-Applikationen können das Fenster-Management an den Tarantella-Client übergeben. Die Anwendung definiert die Größe des Fensters selbst vor. Tarantella-User können diese durch Kleiner- bzw. Größenziehen des Rahmens frei ändern. Dieser Modus läuft nicht mit Betriebssystemoberflächen (KDE, Gnome, CDE, Windows-Desktop), sondern eignet sich nur für einzelne X.11- bzw. Java-Programme (*KWrite*, *xclock*).
- *Independent Windows:* Das Programm läuft in einem eigenem Fenster mit Rahmen in den Größen *Width* und *Height*. Dieser Modus eignet sich für Büroanwendungen wie *kWrite*, *MS Word*, *MS Excel*, *Star Office*. (Beispiel: Abbildung 11.22, Konsole).
- *Kiosk:* Tarantella stellt das Programm ausschließlich auf dem Client ohne Rahmen in der maximalen Größe dar und überdeckt das Betriebssystem des Tarantella-Clients vollständig. Dies empfiehlt sich zur Darstellung von Betriebssystemoberflächen.

Definieren Sie nun wie hier im Beispiel die Applikation *OpenOffice*, und bestätigen Sie Ihre Angaben mit Klick auf die Schaltfläche *Apply*:

Hinweis: Die Suite OpenOffice ist unter Umständen während Ihrer Betriebssysteminstallation nicht mit installiert worden. Sie können für diese Beispiel auch jede andere Applikation verwenden, z. B. `/usr/bin/xterm`.

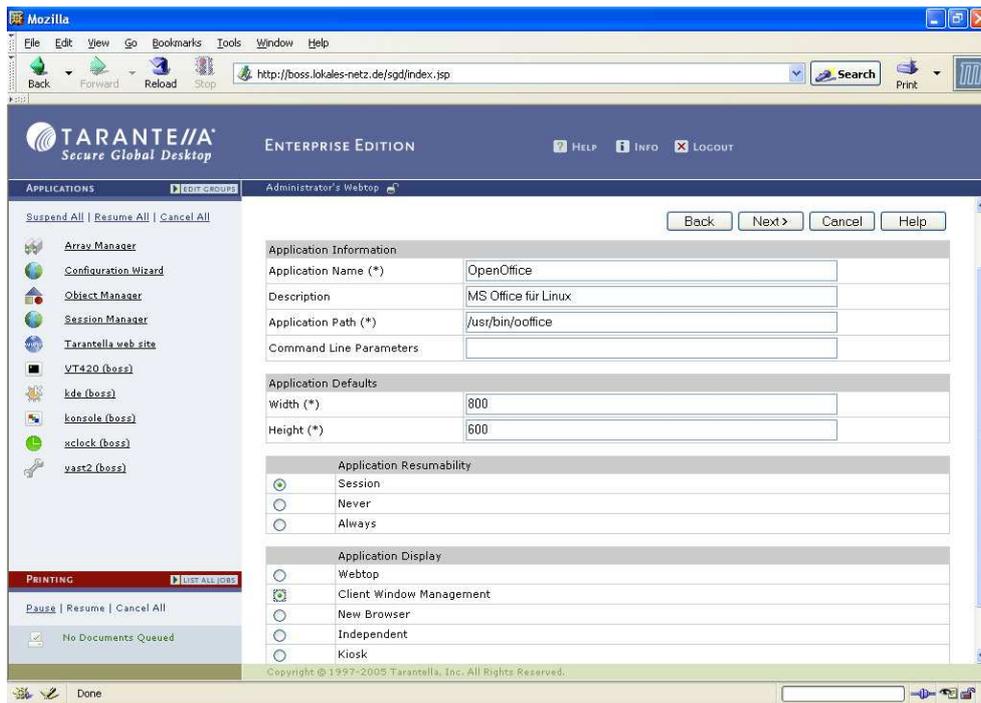


Abbildung 17.18: Neue Applikation OpenOffice, Teil 1

Im zweiten Schritt müssen Sie nun definieren, auf welchem Applikations-Server die Applikation ablaufen soll.

Der *Wizard* bietet Ihnen die zur Verfügung stehenden Server an. Hier wird nun der im ersten Arbeitsschritt angelegte Server angezeigt.

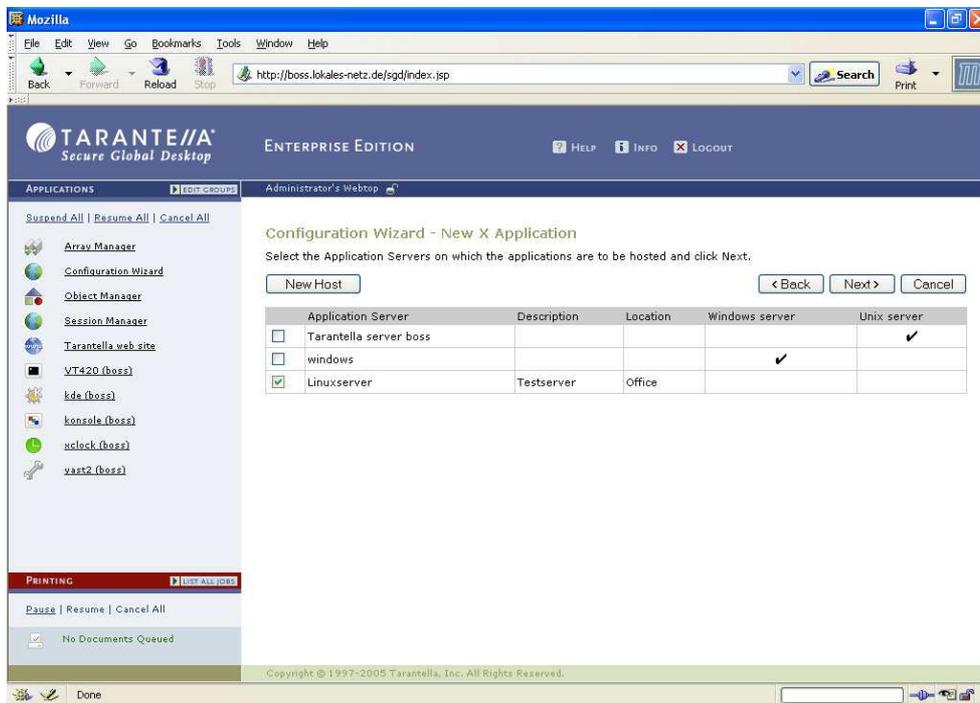


Abbildung 17.19: Neue Applikation OpenOffice, Teil 2

Im letzten Arbeitsschritt muss nun noch definiert werden, welche User bzw. welches User Profile die Applikationen »sehen« bzw. verwenden dürfen. Mit der Auswahl *Unix user Profile* erreichen Sie, dass alle betriebssystemseitigen Anwender OO verwenden können.

Nach einer kurzen Anzeige der Konfiguration können Sie die Angaben mittels Klick auf *Continue* speichern.

Der Benutzer *testuser* kann mit sofortiger Wirkung die Applikation OpenOffice verwenden.

Diese Applikation kann nun via EE auf praktisch jedem Betriebssystem ausgeführt werden. Sie können sich mittels eines Windows-Betriebssystems an EE anmelden und eine X-Applikation ohne einen lokalen X-Server emulieren.

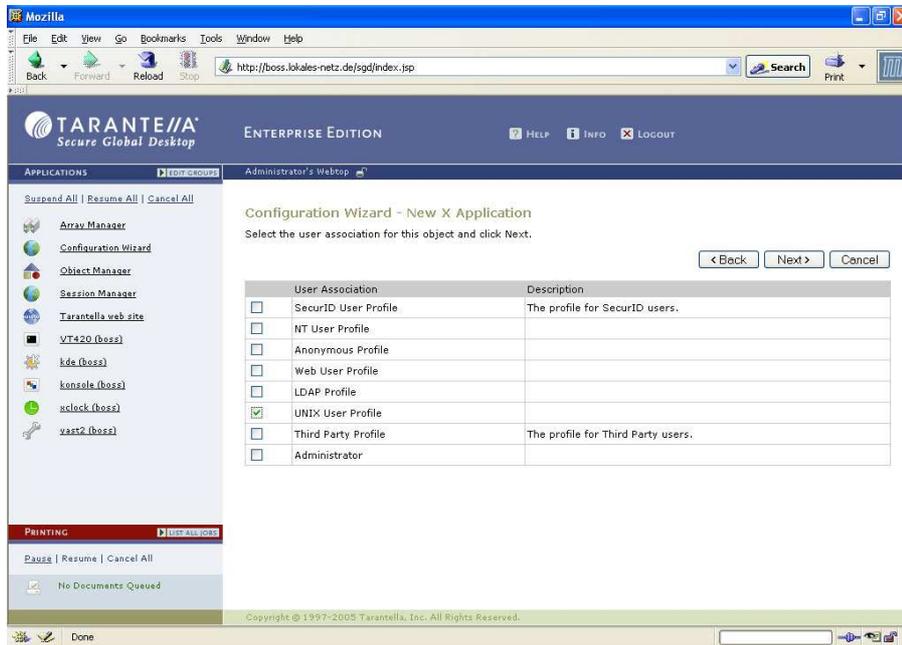


Abbildung 17.20: Neue Applikation OpenOffice, Teil 3

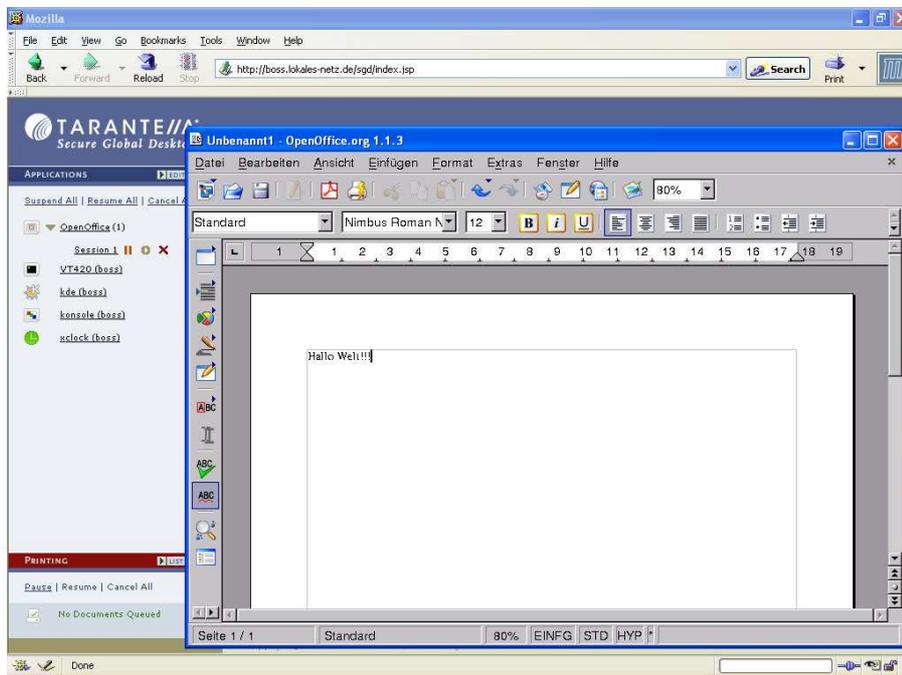


Abbildung 17.21: Applikation OpenOffice

17.4.8 Drucken unter Tarantella SGD EE

Beim Vermitteln von Sitzungen durch EE können Sie im LAN herkömmliche Netzwerkdruckmethoden beibehalten. Wenn Sie jedoch über WAN-Strecken drucken möchten, so empfiehlt sich der Einsatz der Druckdienste von EE. Diese senden die Druckdaten ebenfalls über AIP (bei Bedarf verschlüsselt und komprimiert) an den Standarddrucker des Clients.

EE selbst fungiert hierbei als zusätzlicher Druck-Server, der Druckdaten empfangen und automatisiert an Clients versenden kann.

Der EE-Client erkennt die Client-Drucker automatisiert. Bei einer Anmeldung an EE liest dieser die Drucker des Clients aus und teilt sie dem EE-Server mit.

Möchten Sie von einem Unix/Linux-Applikations-Server drucken, so müssen Sie das Skript `prtinstall.en.sh` von `/opt/tarantella/bin/bin/scripts` auf den Applikations-Server kopieren und dort ausführen. Als einzige Eingabe geben Sie den Namen des EE-Servers an, auf den Anwender drucken sollen. Der eingerichtete Drucker heißt dann `tta_printer`.

Auf Applikations-Servern, auf denen gleichzeitig EE installiert ist, entfällt dieser Schritt.

```
boss:/opt/tarantella/bin/scripts # sh prtinstall.en.sh
DNS name of primary Tarantella server
(press [Return] to accept default [boss.lokale-netz.de]):
Installing printer configuration file...
Installing lp/lpr "wrapper" scripts...
Local system type is Intel Linux.
Informing lp subsystem about local printer tta_printer...
Success.
boss:/ #
```