



OSL Shared Terminal

Collaborative ssh-Client

OSL aktuell - Schöneiche, April 2015

Sie haben Probleme mit einem Linux/Unix-System und benötigen Unterstützung eines externen Dienstleisters oder eines Mitarbeiters, der derzeit nicht vor Ort ist.



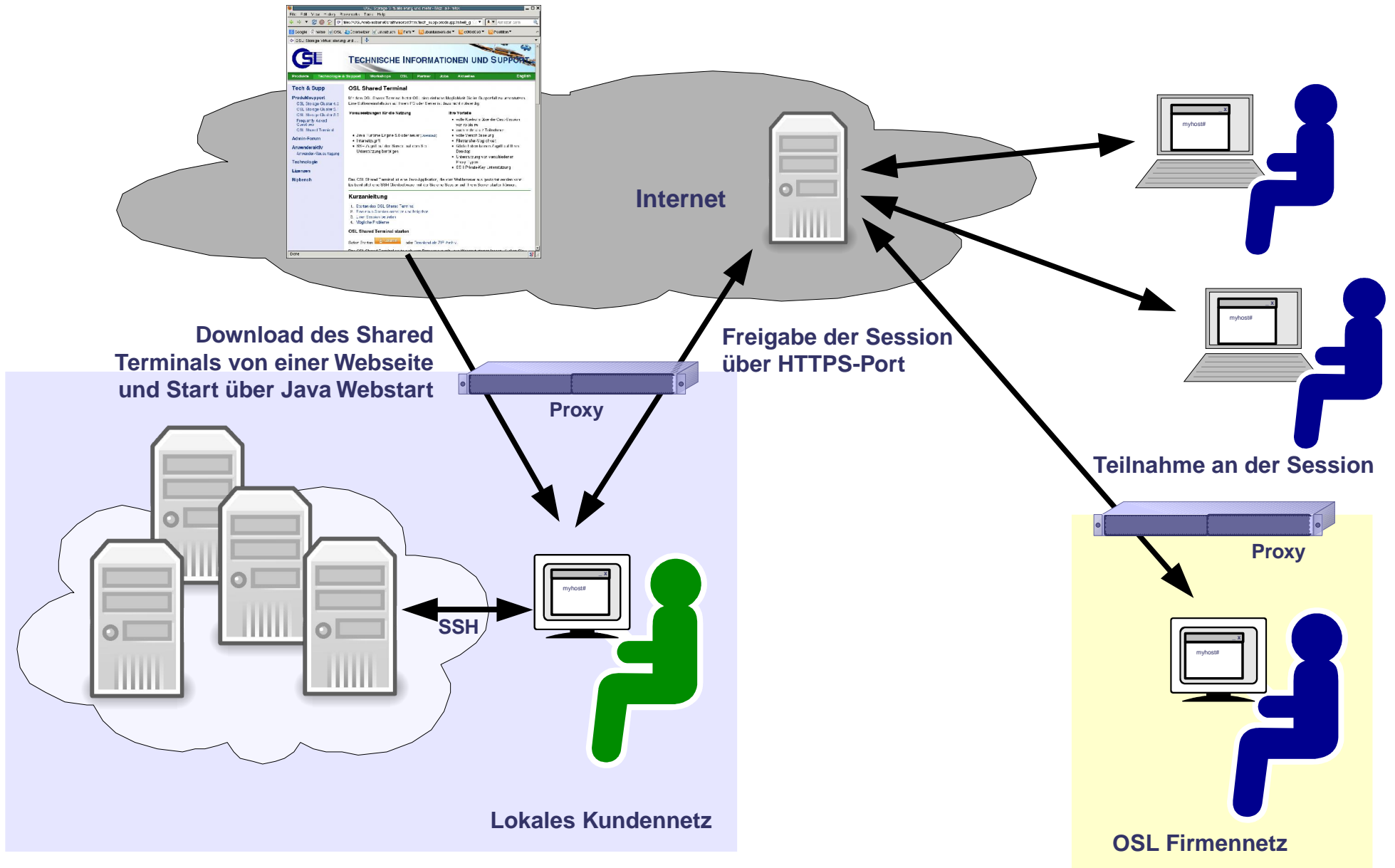
Mögliche Lösungen:

- telefonischer Support
- Einwahl über VPN
- Desktop freigeben (Teamviewer o.ä.)
- SSH Zugang über DMZ (Jumpserver)
- Fernwartungsprogramme

unflexibel, unsicher,
zu komplex, nicht
protokollierbar

OSL Shared Terminal

Funktionsweise





- Java-Applikation – läuft auf allen Betriebssystemen mit JRE 5.0 (oder höher)
- VT100-Terminal mit SSH-Client
- Unterstützung verschiedener HTTP-Proxies (generisch, mit Authentifizierung, NTLM)
- Verkehr zwischen Terminal & Fernwartungsserver ist HTTPS-Traffic (und damit transparent für Firewalls)
- Datenstrom mit AES128 verschlüsselt
- verschlüsselter Datei-Transfer zwischen Session-Teilnehmern und SSH-Server (Upload, Download)
- bis zu 8 Teilnehmer pro Session und unbegrenzt viele parallele Sessions
- Rechtevergabe (nur lesen, lesen und schreiben) für jeden Session-Teilnehmer granular möglich (nur für den Session-Initiator)

OSL Shared Terminal

Vorteile gegenüber anderen Lösungen



- Kein Zugriff auf den Desktop oder den Anwender-PC
- benötigt nur geringe Bandbreite (GSM-Verbindung ausreichend)
- keine Softwareinstallation auf Server oder PC notwendig
- keine Firewall- oder Proxyanpassung notwendig
- Session-Initiator kann die Tätigkeiten verfolgen und protokollieren
- read-only und read-write Zugriff für Gäste
- einzelne Gäste können jederzeit aus der Session entfernt werden
- Passwörter bleiben geheim!
 - Eingaben werden nur an den SSH-Server übertragen
 - angeschlossene Clients sehen nur die Ausgaben des SSH-Servers

OSL Shared Terminal

Shared Terminal in der Praxis



```
OSL Fernwartung - Lokal angemeldet auf: localhost
Session Configure Help
#pragma ident "@(#)netconfig 1.16 99/10/25 SMI"
#
# The "Network Configuration" File.
#
# Each entry is of the form:
#
#     <network_id> <semantics> <flags> <protofamily> <protoname> \
#     <device> <nametoaddr_libs>
#
# The "-" in <nametoaddr_libs> for inet family transports indicates
# redirection to the name service switch policies for "hosts" and
# "services". The "-" may be replaced by nametoaddr libraries that
# comply with the SVr4 specs, in which case the name service switch
# will not be used for netdir_getbyname, netdir_getbyaddr,
# gethostbyname, gethostbyaddr, getservbyname, and getservbyport.
# There are no nametoaddr_libs for the inet family in Solaris anymore.
#
udp6      tpi_clts      v      inet6  udp      /dev/udp6      -
tcp6      tpi_cots_ord  v      inet6  tcp      /dev/tcp6      -
udp       tpi_clts      v      inet   udp      /dev/udp       -
tcp       tpi_cots_ord  v      inet   tcp      /dev/tcp       -
rawip     tpi_raw       -      inet   -        /dev/rawip     -
ticlts    tpi_clts      v      loopback -      /dev/ticlts    straddr.so
ticotsord tpi_cots_ord  v      loopback -      /dev/ticotsord straddr.so
ticots    tpi_cots      v      loopback -      /dev/ticots    straddr.so
~
~
~
"/etc/netconfig" [Read only] 25 lines, 1239 characters
```

Users

- customer (read / write)
- OSLsupp (read only)

SSH connected: schmidt@localhost
Session is shared.
Sessionname: 58618269
Your username is customer



virtualization and clustering – made simple