Remote Desktop Lösungen

Was sie schon immer über remote X wissen wollten

Christian Perle

17. Oktober 2008

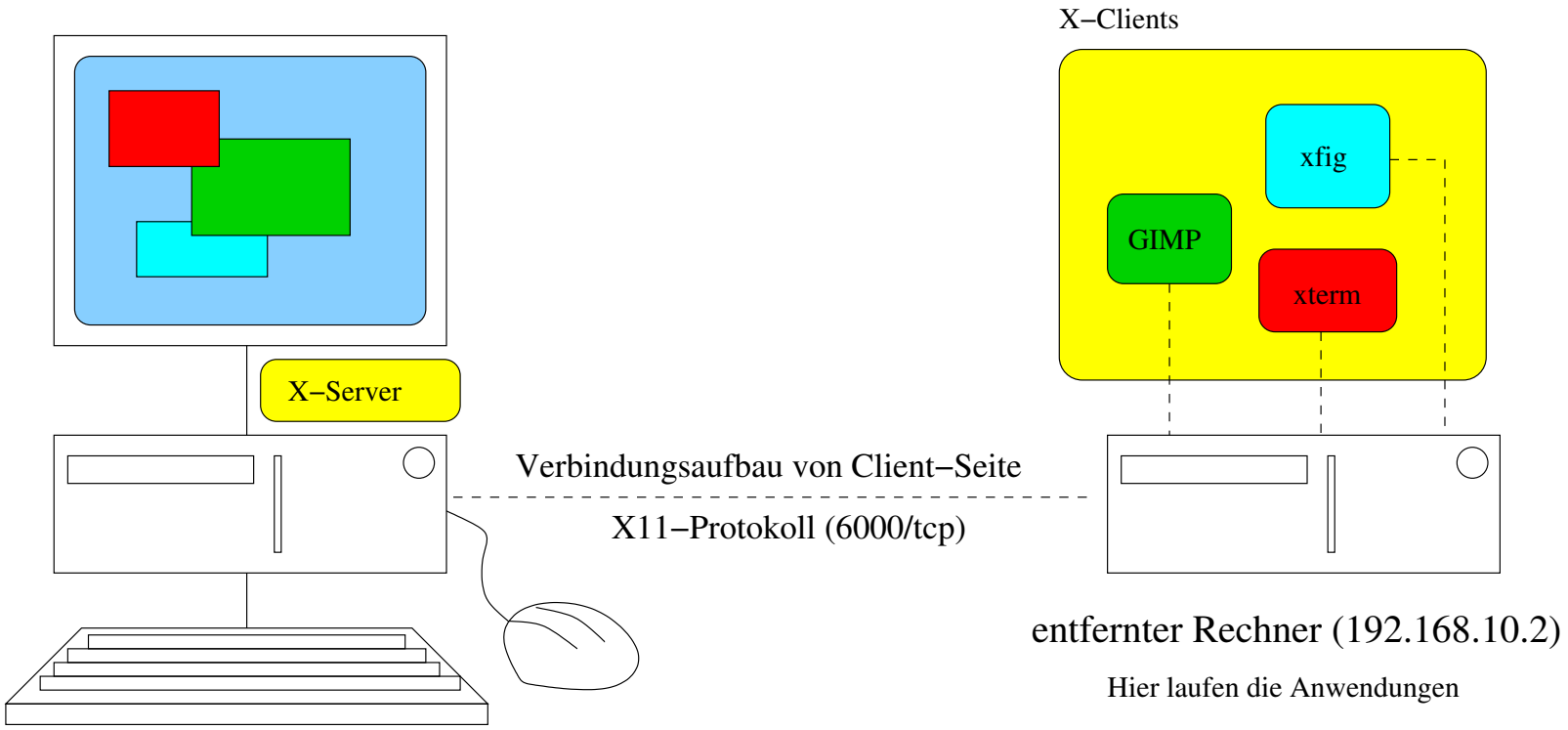
Motivation

- Rechner grafisch aus der Ferne bedienen
- Einzelne Anwendungen
- Kompletter Desktop

X11 ist ein Netzwerkprotokoll

- X ist von sich aus netzwerkfähig
- Begriffe: X-Server, X-Client
- Der X-Server stellt das Display bereit und greift als einziger Prozess auf die Grafikkarte, Tastatur und Maus zu
- Die X-Clients (Anwendungen) verbinden sich über Unix-Domain-Socket (nur lokal) oder über IP mit dem X-Server

Remote X – Verbindungsaufbau



X11 – Displays und Berechtigungen

- Bedeutung der Umgebungsvariable DISPLAY
  Teilt der X-Anwendung mit, zu welchem Display sie sich verbinden soll
  DISPLAY=[hostname]:displaynummer[.screennummer]
- Einzelne Anwendungen umleiten

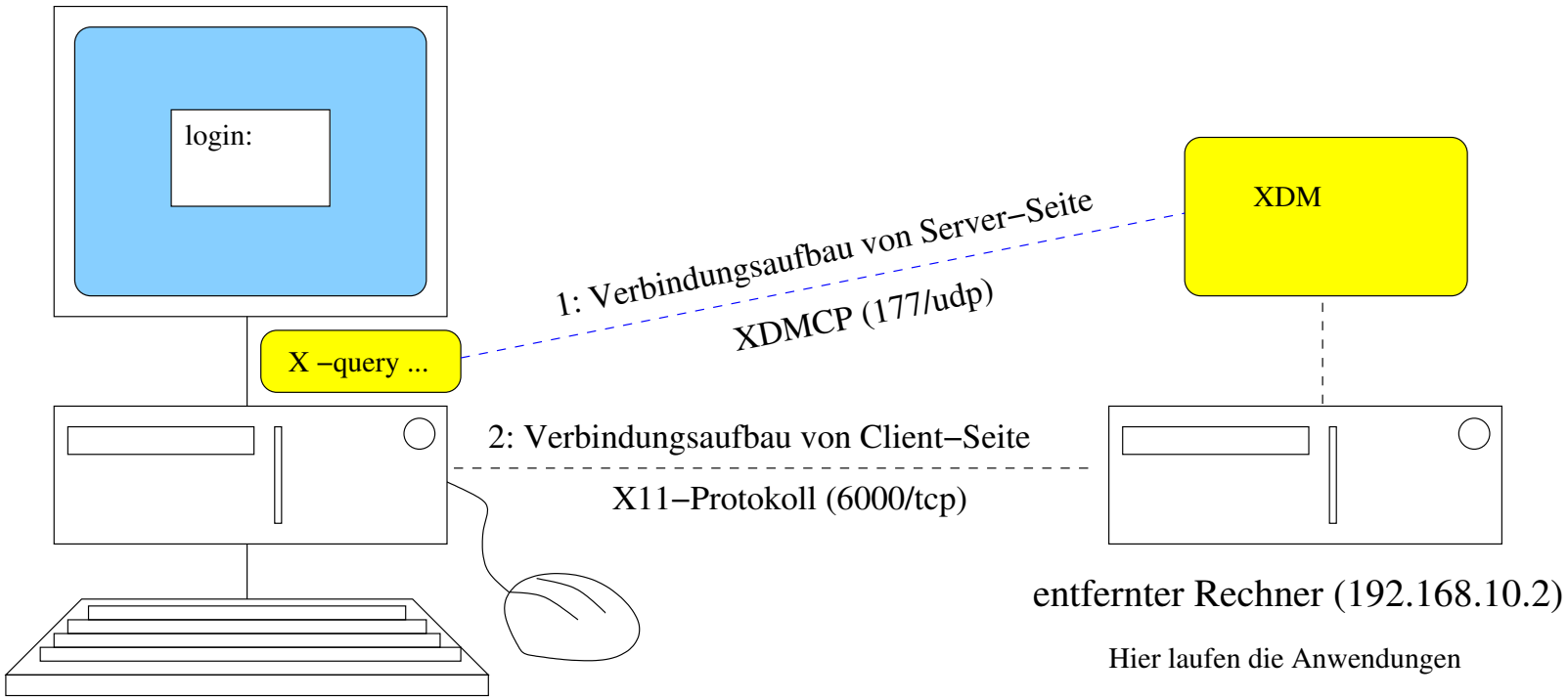
```
[ich@192.168.10.1]$ ssh ubuntu@192.168.10.2
[ubuntu@192.168.10.2]$ export DISPLAY=192.168.10.1:0
[ubuntu@192.168.10.2]$ gwenview &
[ubuntu@192.168.10.2]$ konqueror &
```

X11 – Displays und Berechtigungen

- Display-Berechtigungskonzept
  - Berechtigung auf Host-Basis: `xhost`
  - Berechtigung mit Cookies: `xauth`
- Kompletter Remote Desktop inklusive grafischem Login (Display Manager): xdm/gdm/kdm

```
[ich@192.168.10.1]$ sudo Xorg vt8 :1 -query 192.168.10.2
```

XDM-Rollentausch: Erst Server, dann Client



X11 – Schwächen

- X11 ist nicht NAT-fähig
  (Rückverbindung von remote zu lokal schlägt fehl)
- XDMCP hat zusätzliche Probleme mit NAT
  (DISPLAY wird „falsch“ gesetzt)
- Unverschlüsselt
- Kommt schlecht mit hoher Latenz im Netz zurecht

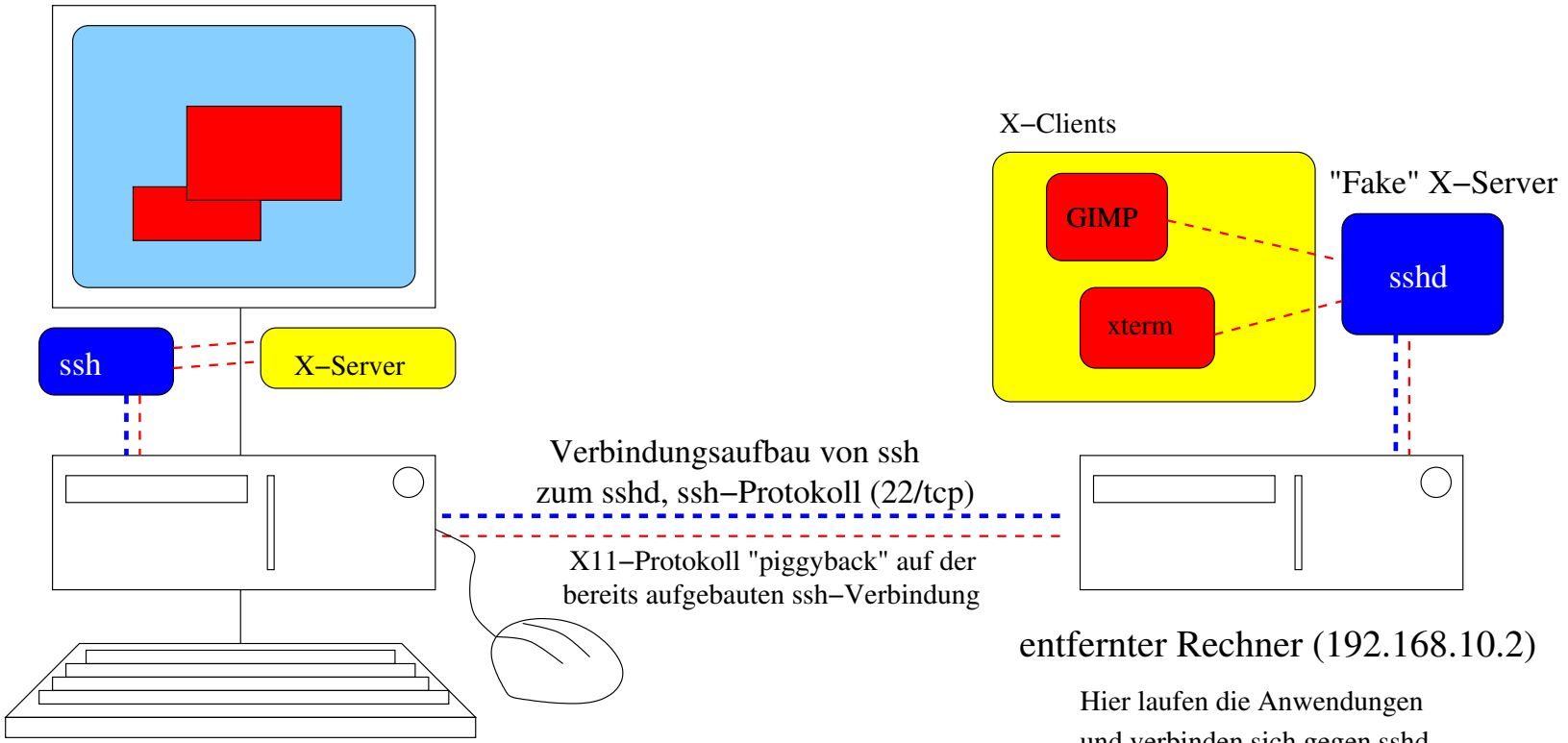
Secure Shell als X-Weiterleitung

- Unterschiede zu Remote X
  (Tunnel durch existierende ssh-Verbindung)
- Leitet nur einzelne X-Clients um

```
[ich@192.168.10.1]$ ssh -X ubuntu@192.168.10.2
[ubuntu@192.168.10.2]$ kate &
[ubuntu@192.168.10.2]$ xeyes &
```

- NAT-fähig
- Verschlüsselt
- Setzt Berechtigungen mit xauth automatisch
- Kommt schlecht mit hoher Latenz im Netz zurecht

Secure Shell als X-Weiterleitung (Tunnel)



lokaler Rechner (192.168.10.1)

Hier werden die Anwendungen dargestellt
und hier bekommen sie ihre Eingabe

entfernter Rechner (192.168.10.2)

Hier laufen die Anwendungen
und verbinden sich gegen sshd

Virtual Network Computing

- Immer kompletter Desktop
- Sitzung kann „geparkt“ werden
- Ermöglicht shared- und readonly-Verbindungen
- Separater Desktop (Xvnc)

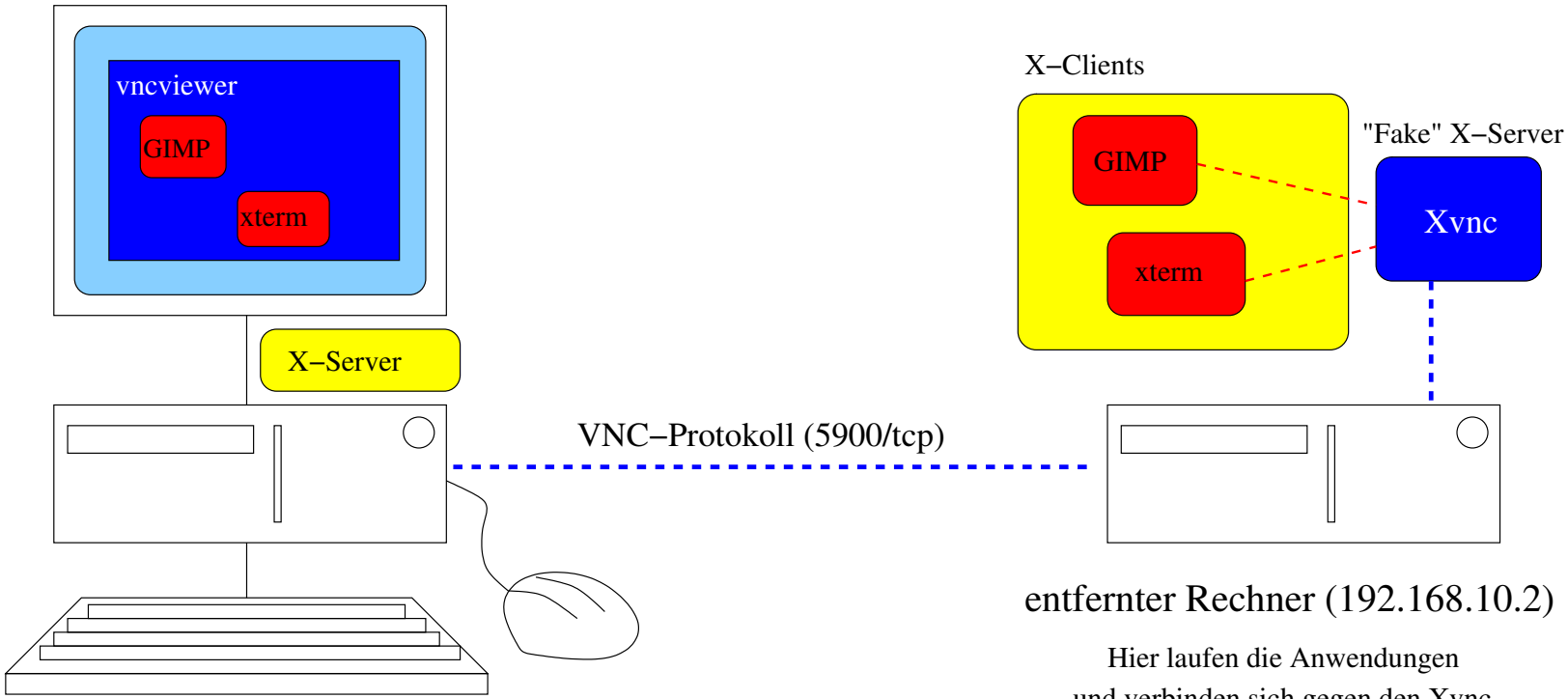
```
[ubuntu@192.168.10.2]$ tightvncserver :1 -geometry 800x600
```

- Existierender Desktop (x11vnc)

```
[ubuntu@192.168.10.2]$ x11vnc -shared -readonly
```

- In Gnome (vino) und KDE (krfb) bereits integriert

Virtual Network Computing, separater Desktop



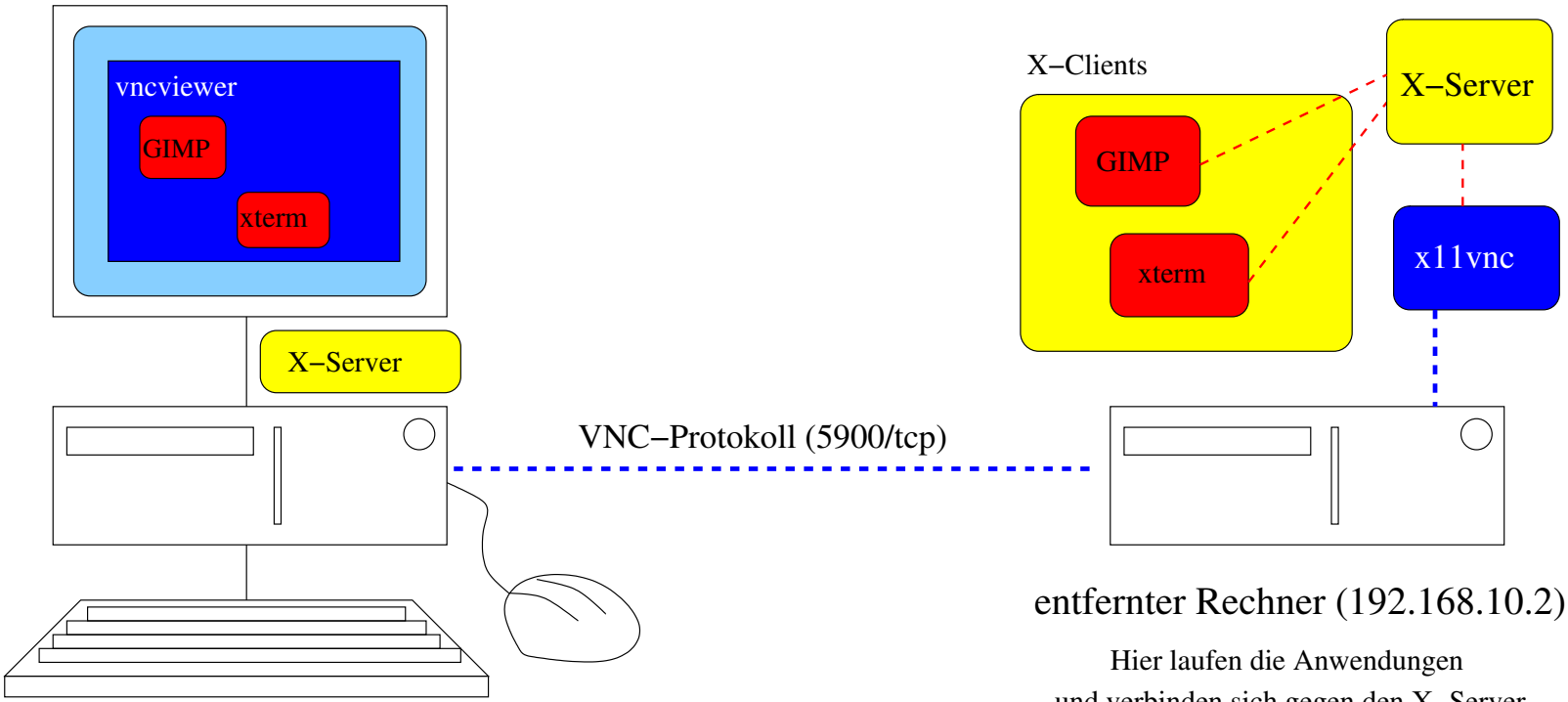
lokaler Rechner (192.168.10.1)

Darstellung des entfernten Desktop
im vncviewer–Fenster

entfernter Rechner (192.168.10.2)

Hier laufen die Anwendungen
und verbinden sich gegen den Xvnc

Virtual Network Computing, existierender Desktop



lokaler Rechner (192.168.10.1)

Darstellung des entfernten Desktop
im vncviewer–Fenster

entfernter Rechner (192.168.10.2)

Hier laufen die Anwendungen
und verbinden sich gegen den X–Server

x11vnc ist auch ein X–Client!

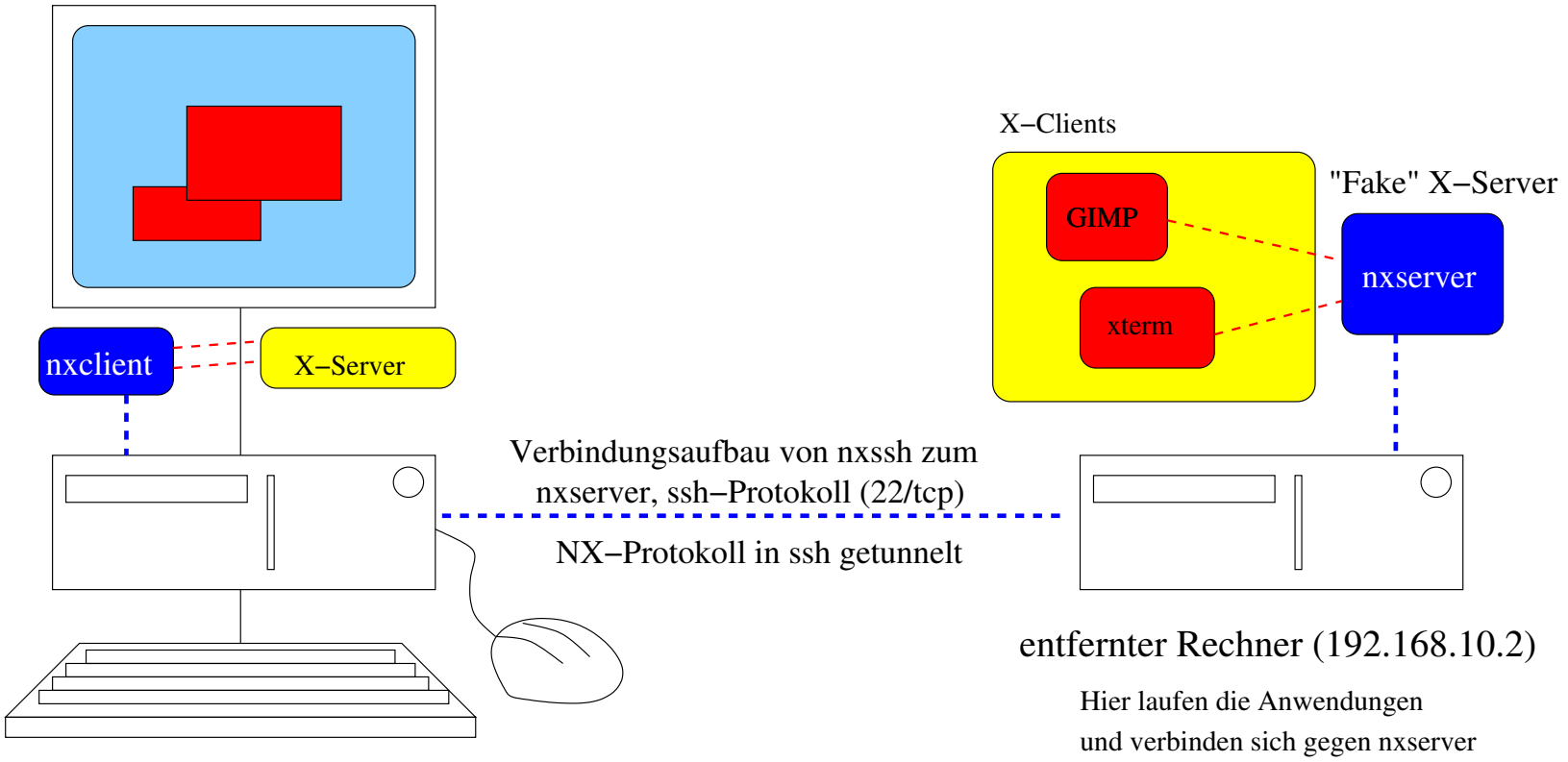
VNC – Schwächen

- *Sehr* bandbreitenhungrig, nur sinnvoll im LAN nutzbar
- Unverschlüsselt
- Keine vollständige Sitzungsverwaltung

NoMachine NX

- Starke Optimierung des X11-Protokolls (Protokoll-Proxy)
- Kompletter Desktop oder einzelne Anwendungen
- Separater oder existierender Desktop (Shadow)
- Relativ immun gegen hohe Latenz und geringe Bandbreite
- Sitzungsverwaltung
- Sitzungen können „geparkt“ werden
- Verschlüsselt

NoMachine NX



NX – Schwächen

- Closed source
- Serverseite auch quelloffen verfügbar (FreeNX)
- Keine shared- und readonly-Verbindungen möglich

Aufgaben

1. GDM für Remote-Zugriff (XDMCP) konfigurieren
2. Fluxbox in NX-Session
3. Zugriff auf Xvnc-Server über ssh tunneln
4. „Big Brother Desktop“