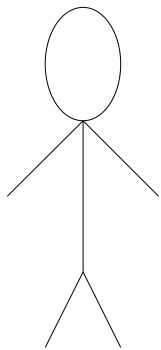
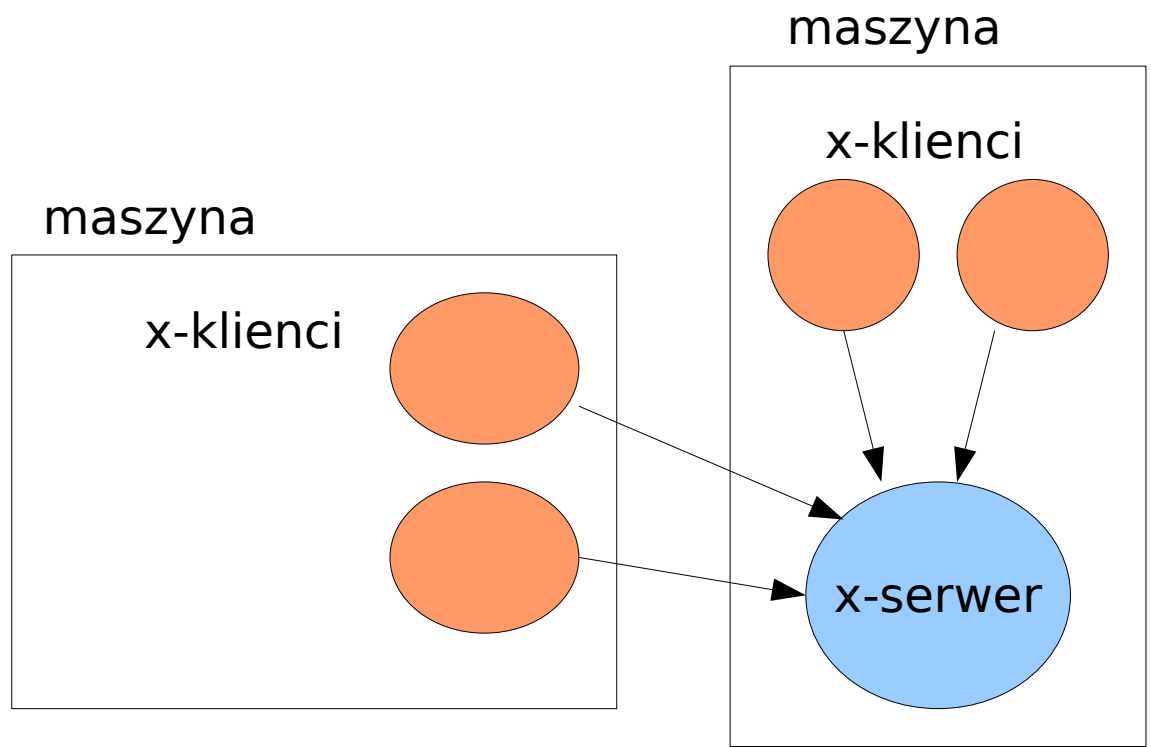
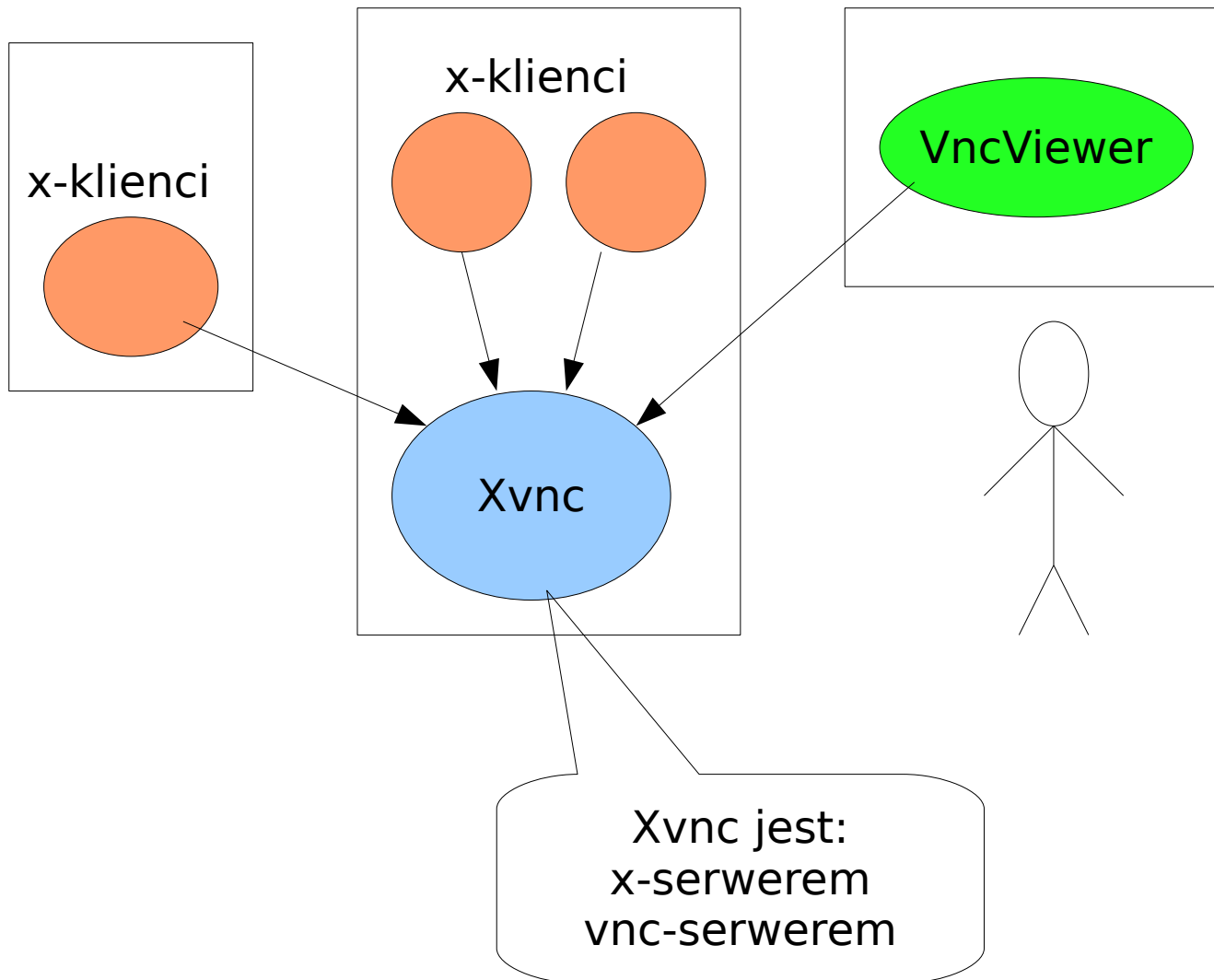


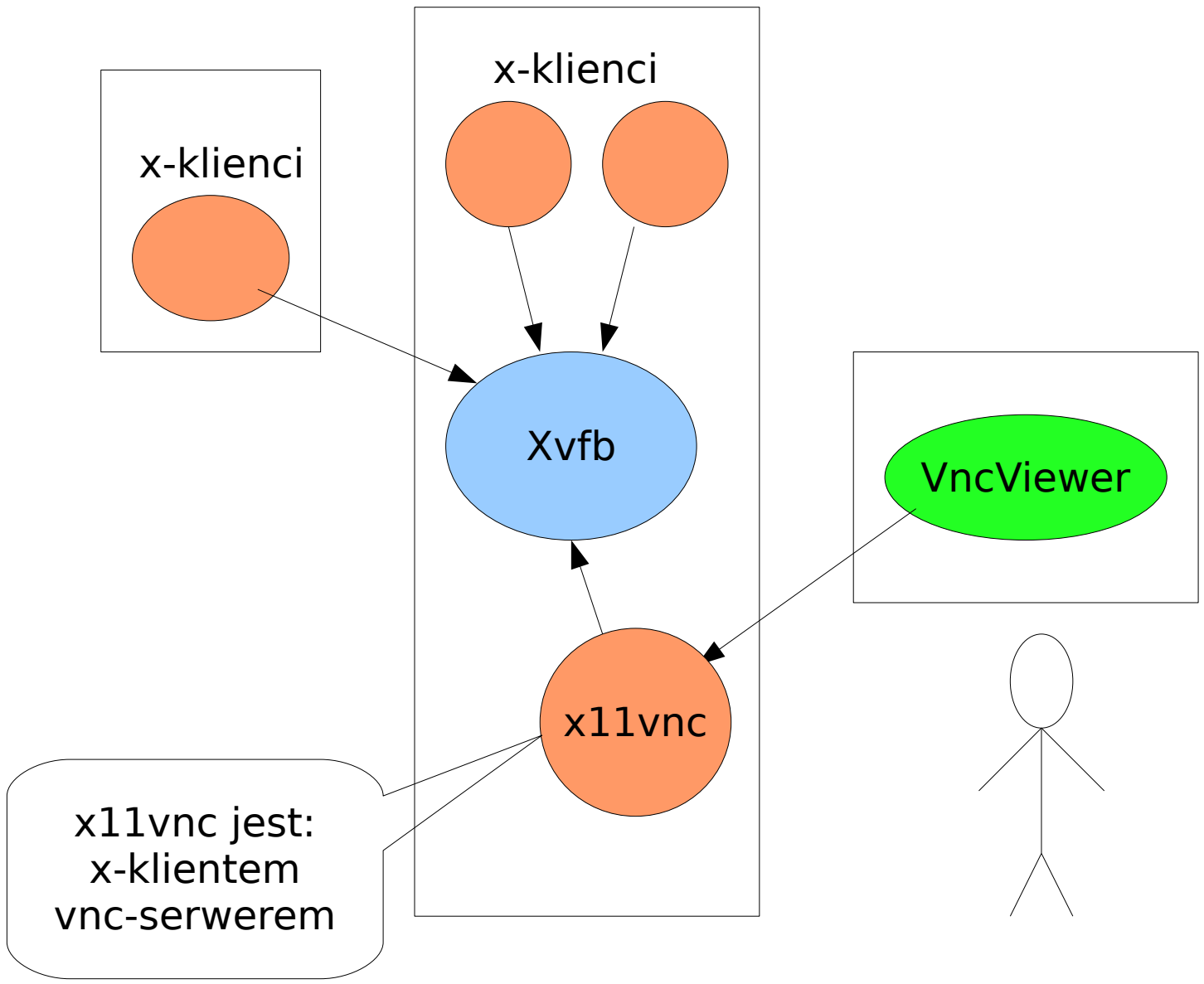
# X Windows

- x-serwer (X/Xorg, obsługa ekranu, klaw./myszy)
- x-klienci (programy okienkowe)
- Środowisko desktopowe (KDE, GNOME, ...)
- Window Manager, \*wm, np. jwm  
co jeszcze jest w środowisku desk. ?  
Odp: toolkity (bibl .so), programy, pliki konfig, ...
- Inne prog: Xvfb, Xvnc, x11vnc
- VNC – coś w rodzaju „zdalnego desktopu”
- Skąd x-kli wiedzą do którego x-ser się podłączyć?  
`export DISPLAY=host:display_num.screen_num`  
lub opcja `-display "..."`

- Uruchamianie x-serwerów:  
`Xvfb :1 -ac -screen 0 700x500x24 &`
- Zabezpieczenia:  
`xhost +/-host`; na poziomie hosta  
`xauth ...` bardziej zaawansowane  
`vnc ... hasło, vncpasswd`
- Zasoby X (ang. resources), plik `.Xdefaults`,  
wartości domniemane opcji/właściwości  
`tclkit*Button.background: red`
- Fonty X (czcionki), 2 rodzaje:  
„stare”/core fonts vs „nowe”/Xft fonts:  
freetype, aa, lepiej wyglądają, x-kli je rysuje a nie x-ser...
- Rozszerzenia X-ser:  
GLX, grafika 3D (OplenGL) w oknie X, akceleracja GPU  
RANDR, polecenie `xrandr`, konfiguracja „na żywo”  
RENDER, używane przez fonty Xft
- Konfiguracja statyczna X-ser: `/etc/X11/xorg.conf`







# Programowanie pod X

- Programowanie niskopoz.: bibl. Xlib  
Toolkity z widgetami: Qt, GTK+, Tk (Tcl/Tk)
- Widget: okno + proc. rysująca wewnątrz okna
- widget zawiera inne widgety...  
*drzewko widgetów! ścieżki z kropkami!*  
*jak są ułożone? geometry manager!*
- Przykłady widgetów Tk: *toplevel, frame, label, button, entry, listbox, scrollbar, canvas, text*
- Nie tylko j. Tcl: wszystkie j. skryptowe używają Tk:  
python, ruby, perl, ... patrz: <http://www.tkdocs.com/>
- Tcl/Tk przykład:  

```
label .et1 -text „a ku ku !”  
pack .et1 -side left
```