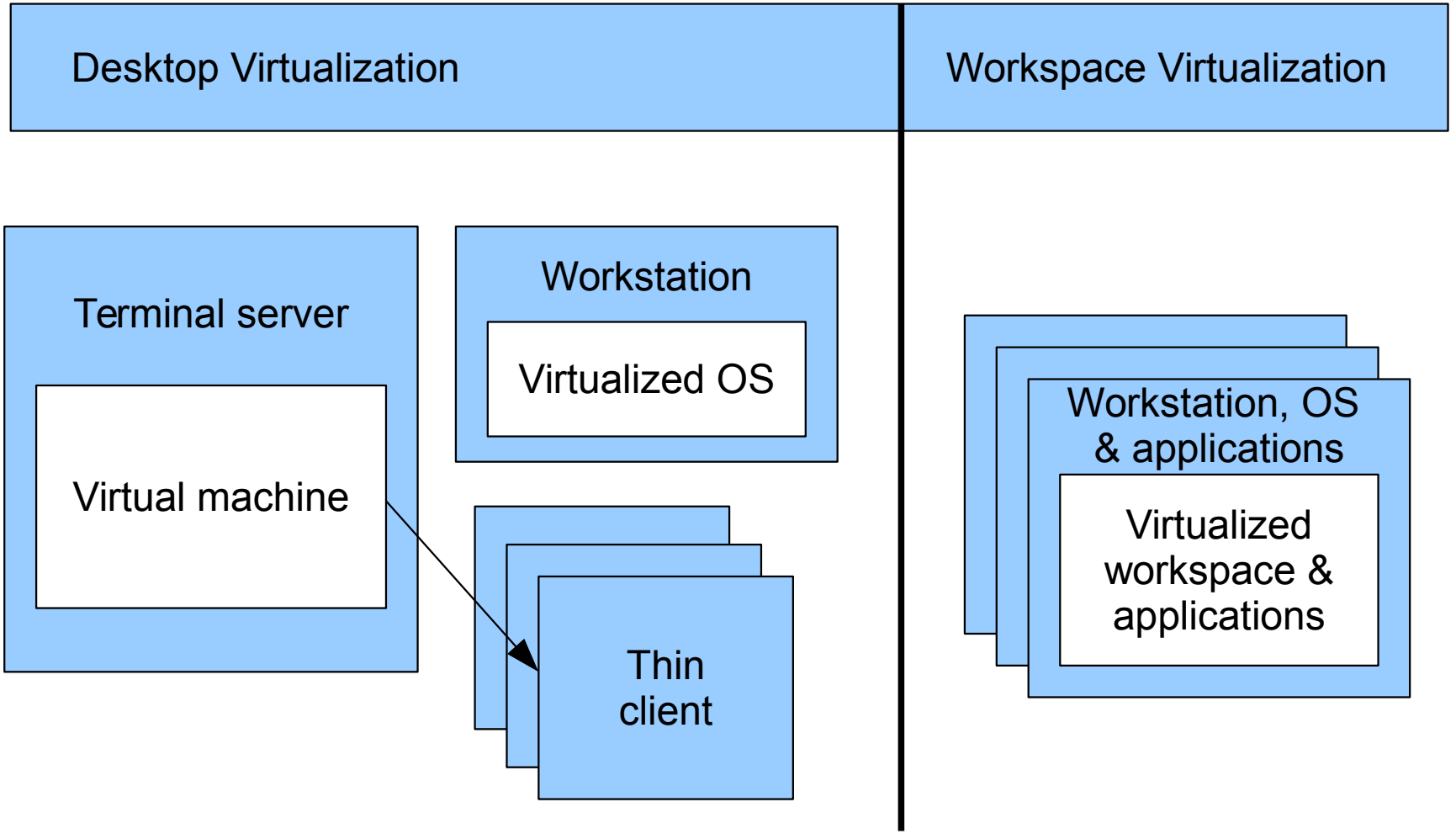
Scaling down insecure desktop operating systems to virtualized desktop environments

Dmitriy Kostiuk

Brest State Technical University
dmitriykostiuk@gmail.com

Virtualization on the desktop



When thin clients have no sense

- You already own enough workstations
- Thin price gap between office workstation and thin client
- Using old workstations as thin clients is not the best choice for your funding model
- Additional network load

Why do we like VM on desktop?

- Guest OS is abstracted from the computer hardware diversity
- External secutiy hazards have less influence on the virtualized environment
- Immediate restore after falures can be easily automated (snapshots)
- Fast deployment & easy replication of the ready-to-use workspace

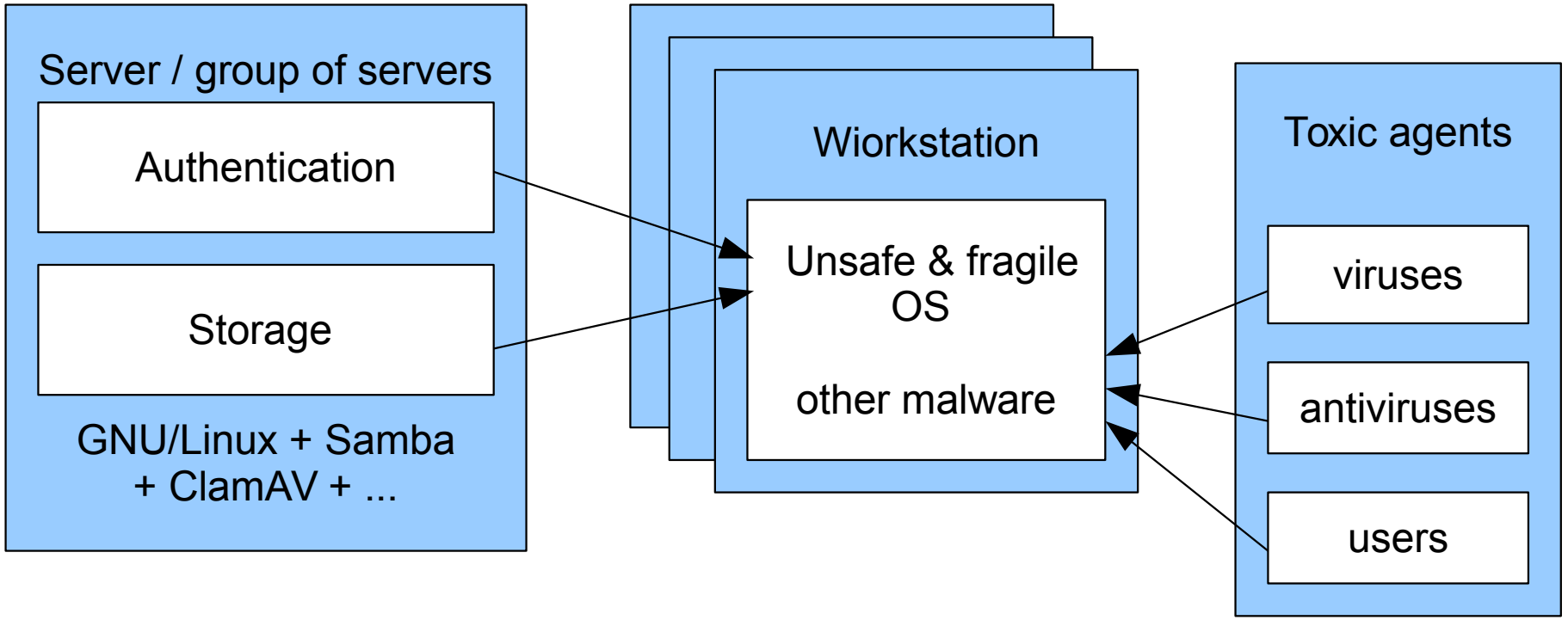
Projects with approach similar to workspace virtualization

- Container-style virtualization
  - Bedrock Linux (meta-distro which strips down other Linux distributions and allows user to run them chrooted)
  - Qubes OS (Xen-based distro with set of apps in separate isolated domains)
- Virtualization-less workspace wirtualization :)
  - Some admin tools treating «workspace» as bundle of application to be deployed on workstation

Transparent virtualization

- Unmodified OS is running in a virtual machine sandbox
  - VirtualBox, QEMU
- User does not see virtualization software
  - isimilar to workspace virtualization
- Some activities (login, volumes mount, etc.) are delegated to the host OS
  - Resources are verified and forwarded by host OS and just appear in guest OS in a magical manner

Without virtualization



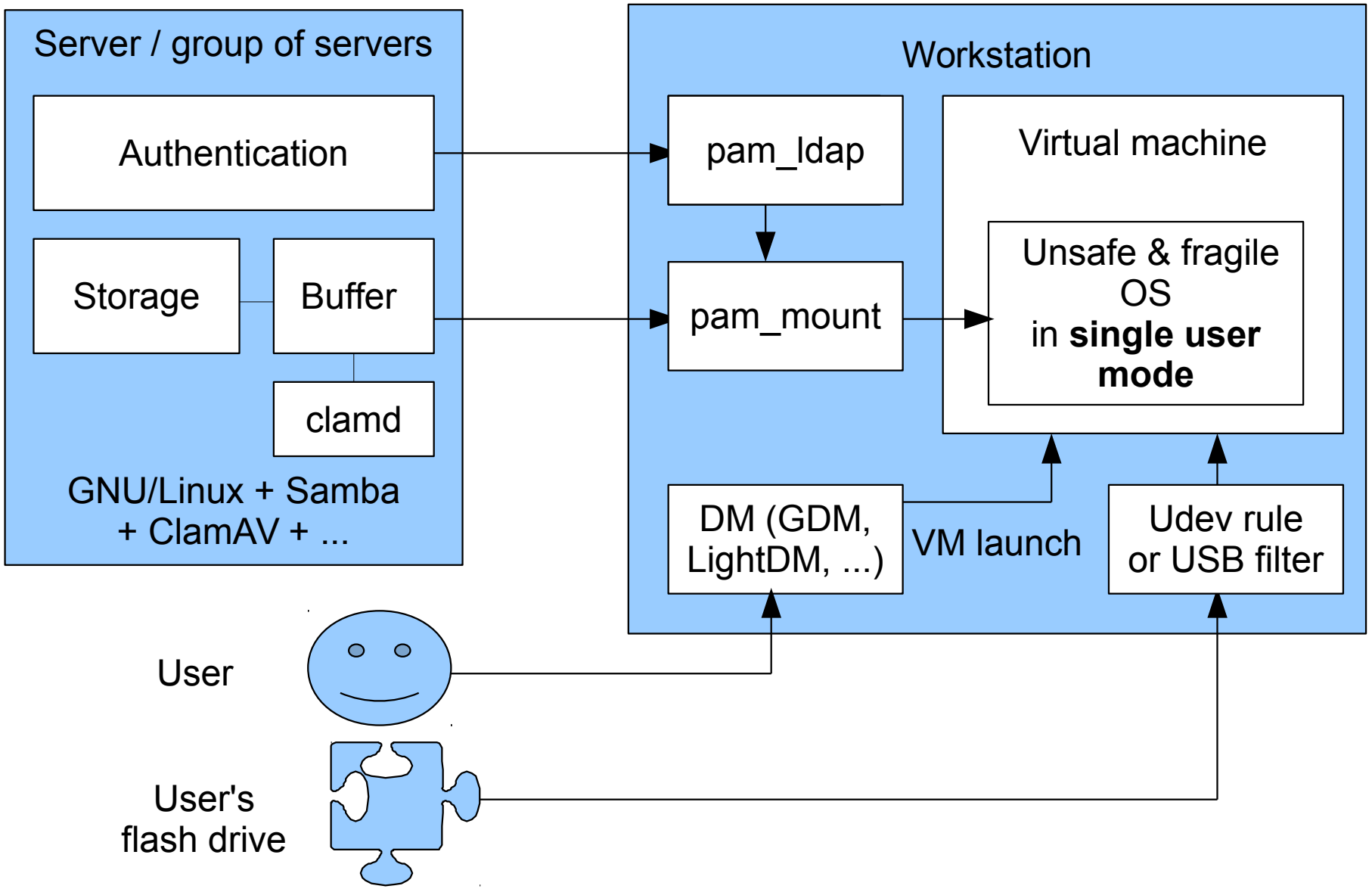
Example: Microsoft OS in a sanbox

- No need in failure recovery
  - Snapshots provide fresh system on each boot
- No need in antivirus monitor on workstation
  - Just relaunch (relogin) it to repair
- Out-of-workflow network activity is more difficult
  - VMs use NAT networking by default
- Single image can be used at all workstations
  - Forget about tricky tuning for domain joining, etc.

Choice of virtual machine

Feature	VirtualBox	QEMU
Accelerated graphics	Yes	2D for xorg; 3D not ready yet
High performance	Yes (esp. with guest additions)	Yes (esp. with virtio guest drivers)
USB forwarding	Yes (filter by device or socket)	Yes (filter by device)
Networking & shared folders	Yes & Yes	Yes & Yes
Fixed drives and RAM snapshots	Yes (drive or RAM but not both?)	Yes

Example: sandboxing Microsoft OS



Virtualbox scenario

- Configure GNU/Linux workstation
  - Authentication via pam_ldap
  - Network storage mounted via pam_mount
- Configure VirtualBox for multi-user usage
  - move «.VirtualBox/» to some common area (/var)
  - Symlink it to the home folder temlate (/etc/skel/)
- Prepare VM image with additional steps
  - Fix disk image to make it restore at reboot
    - VBoxManage modifyhd --type immutable …...
  - Configure USB forwarding

VM launcher with shortcut in /usr/share/xsessions

- Launches sandbox (VBoxManage)
- Tunnels already mounted network storages into already launched VM via Shared Folders
- Waits until VM finishes

```
#!/bin/sh
vm="$1"
disp="$2"
home="$3"
export DISPLAY=$disp
export VBOX_USER_HOME="/var/virt/vbox/home/"
/bin/sh -c "sleep 3;VBoxManage sharedfolder add ${vm} --transient --name private
    --hostpath $home/first_mounted_share" &
/bin/sh -c "sleep 5;VBoxManage sharedfolder add ${vm} --transient --name public
   --hostpath $home/second_mounted_share" &
VBoxSDL --fullscreen --fullscreenresize --startvm ${vm}
```

VirtualBox vs. QEMU: snapshots

- VirtualBox
  - Make disk image immutable via VBoxManage
  - Inoncompartible with RAM snapshots, so guest OS will boot at every launch
- QEMU
  - Create base image with OS and applications, make derived image and save RAM snapshot into it
    - qemu-img create -f qcow2 -o backing_file=base.img derived.img
    - qemu-system-i386 -hda derived.img …
  - Backup derived image, schedule its daily restore...

VirtualBox vs. QEMU: acceleration

- VirtualBox
  - Install guest additions if present. Check 2D & 3D acceleration in VM properites
- QEMU
  - Install guest drivers for virtio block and/or network devices, if present (virtio-gpu is not 3D-capable yet)
  - Add virtio parameters to qemu launcher
    - qemu-system-i386 -boot order=c -drive file=disk.img,if=virtio ...
    - qemu-system-i386 -net nic,model=virtio ...

Without guest drivers

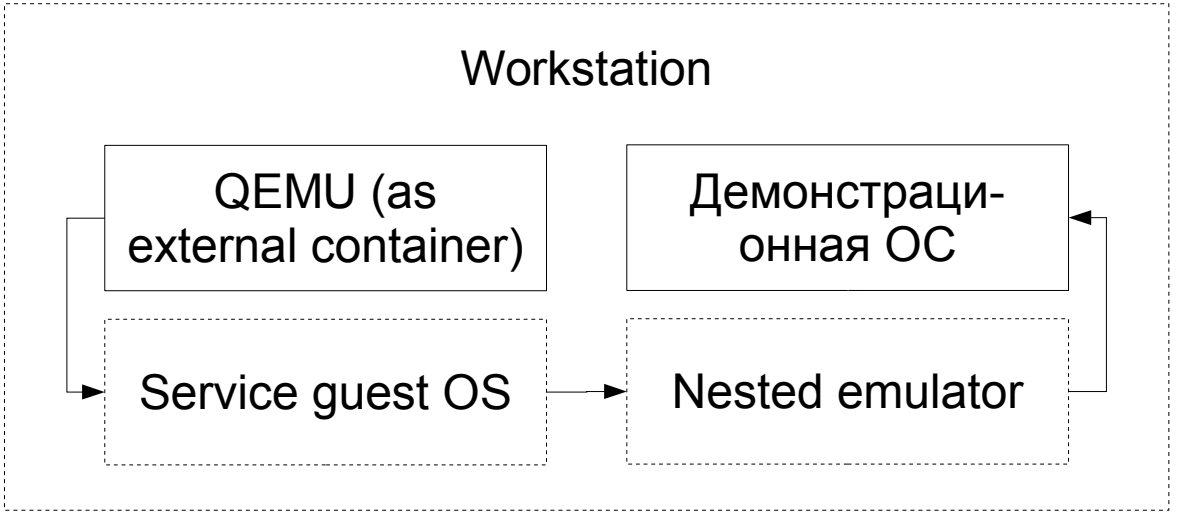
- If there are no guest additions / drivers available, supply guest OS with drivers for…
- Wacom USB tablet …
  - it worth a separate slide :)
- ...and SMB network
  - QEMU provides shared folders via embedded smb server
  - VirtualBox at least allows to do the same

A separate slide :)

- Emulated mouse operates with relative coordinates
- Host OS gives you absolute coordinates…
  - ...and knows nothing about mouse pointer acceleration added by guest OS
  - Therefore in reality you have two cursors in different positions, moving with different speed
  - User doesn't notice it when host cursor is hidden :)
- Any absolute coordinates pointing device saves the situation

Nested virtualization

- QEMU acts as a container for OS snapshots
  - It runs service guest OS, which itself runs an emulator...
    - ...which runs your workspace :)



Possible service operating systems:

- **Linux**
- **FreeDOS**
- **ReactOS**

Results

- Advantages
  - User has administrator priviledges inside of unnoticeable sandbox
  - «Virtualize and forget» principle
- Disadvantages
  - There's no automatic solution
  - Needs some administration skills
  - Some person would come one day to comment that placing VM shortchut on the user's desktop is much simpler :)))