

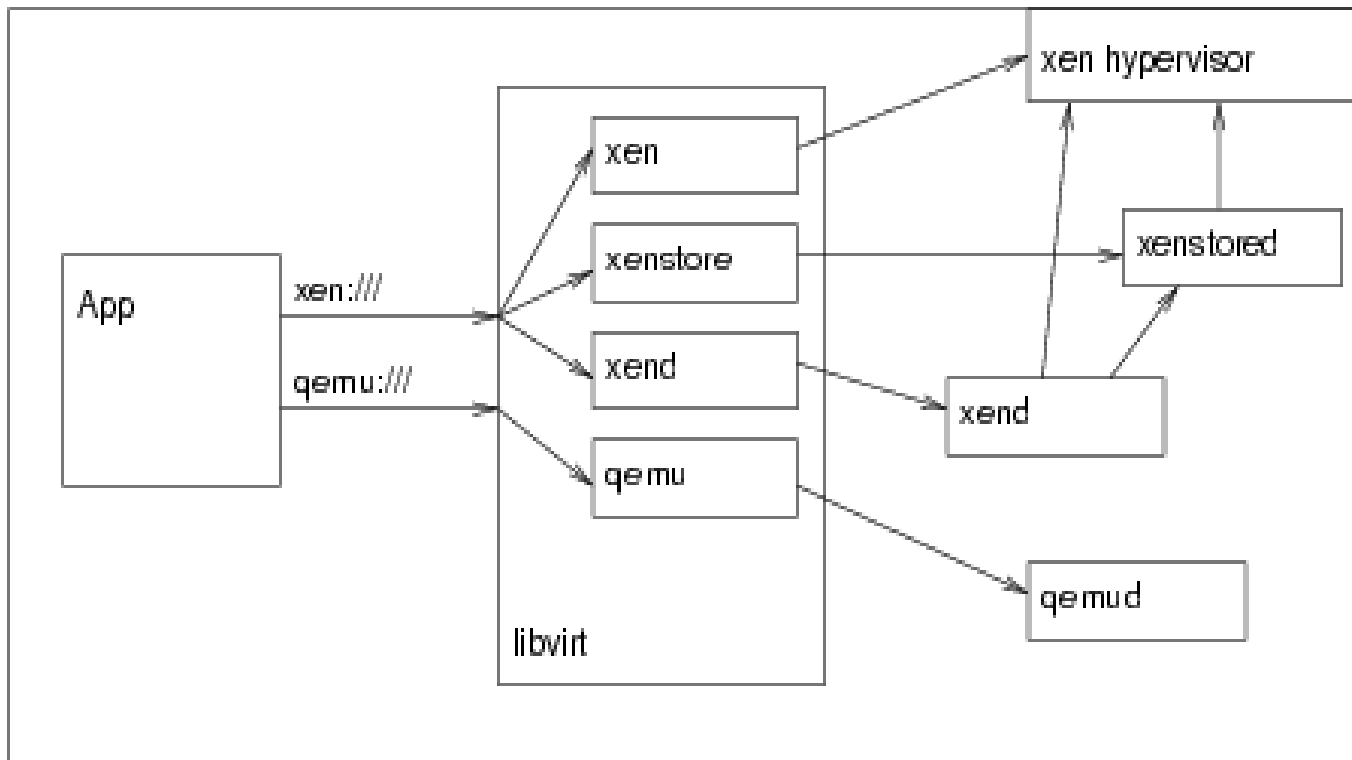
Secure remote management with virtualization

Daniel P. Berrangé <berrange@redhat.com>

# libvirt: Background

- API for management of hypervisors
- Community (Red Hat, Fujitsu, Bull)
- Isolates apps from HV specific APIs
- Driver support for Xen, QEMU, KVM
- C, Python, Perl, shell APIs (`virsh`)

# libvirt: Local Architecture



# libvirt: Remote Management

- Local management unchanged
- Driver talks to remote libvirtd server
- XDR messaging protocol (rfc 1832)
- Layered over TLS 1.1 or tunnel SSH
- x509 certificate authentication
- Role based MAC with SELinux



# libvirt: Host Capabilities

- Supported architectures: x86, ppc, sparc
- Supported virt types: Xen, KVM, QEMU, KQEMU
- Supported OS types: Xen PV, HVM
- CPU capabilities: SVM, VMX, PAE

# libvirt: Network Management

- Shared physical device / virtual network
- APIs to define virtual networks
- dnsmasq provides DHCP + DNS
- Isolated or NAT forwarding (iptables)
- Solve NetworkManager/Laptop case

# libvirt: Storage Management

- Storage pool of file, partition, or lvm
- Enumeration volumes in pool
- Allocate virtual disks from pool
- Verify availability for migration
- POSIX (file), GpartD (partition), ??? (lvm)

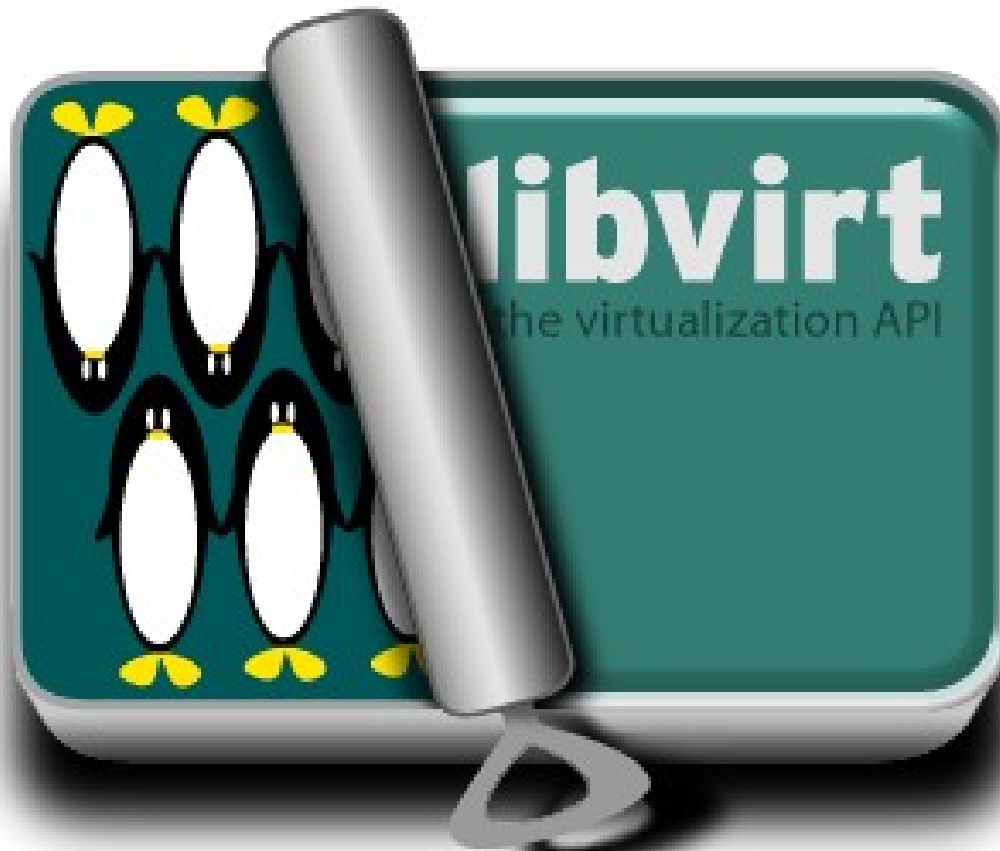


# libvirt: Graphics Console

- Xen, QEMU, KVM provide VNC server
- VNC unencrypted traffic, 'trivial' auth
- Goal for parity auth scheme with libvirt
- VeNCrypt extension adds TLS + x509
- Port PV daemon to use QEMU VNC code
- GTK-VNC client supports VeNCrypt

# libvirt: Text Console

- Xen, QEMU, KVM provide Pseudo-TTY
- Restricted to root on local machine
- QEMU provides UNIX/TCP socket access
- Goal for parity auth scheme with libvirt
- Existing tool ? Tunnel VNC / libvirt ?



<http://libvirt.org/>