



Informatik der Systeme – Chapter 9: Virtualization

Prof. Dr. Michael Menth

<https://kn.inf.uni-tuebingen.de>



- ▶ A software implementation of a machine (i.e., a computer) that executes programs like a physical machine
- ▶ Software running inside a VM is limited to the resources and abstractions provided by the VM
 - it cannot break out of its virtual environment

(Source: Wikipedia)

- ▶ Two categories
 - Process (or application) VM
 - System (or full) VM



- ▶ Designed to run a single program, i.e., it supports a single process
- ▶ Provides a platform-independent programming environment
 - Abstracts away details of the underlying hardware or operating system
 - Allows a program to execute in the same way on any platform
- ▶ Examples
 - Java / Java VM
 - .NET / Common Language Runtime

(Source: Wikipedia)



- ▶ Objective: sharing of resources
- ▶ General concept
 - Multiple virtual machines (VMs) simultaneously run on one physical host machine
 - Host computer emulates machine resources and guest operating system(s)
- ▶ VMs
 - Provide a complete system platform which supports the execution of a complete operating system (OS)
 - Qualitatively behave like physical machines
 - Are isolated against each other, i.e., cannot damage each other or the host machine
- ▶ Advantage over multi-user OS
 - Improved isolation of multiple guests
 - Reduces security and reliability problems



► Resource sharing saves resources

- Capital expenditure (CAPEX)
 - Mostly hardware
 - Compute nodes
 - Networking equipment
- Operational expenditure (OPEX)
 - Space
 - Energy
 - Cooling
 - Possibly administration

► More flexibility

- VMs can be „migrated“ from one machine to another, practical for maintenance purposes
- VMs faster to set up than a physical machine, practical for experimental purposes

► Increased security

- Resources for a VM (CPU, RAM, disk, number of processes) can be limited
- Run web servers on VM $\Rightarrow$ host system is not compromised
 - If web server faces denial of service attack (DoS)
 - If web server is hacked

► Efficient use of physical resources

- Services typically provided on dedicated physical machines, low resource utilization
- May be provided on VMs on a single host
- Data centers run VMs of different clients on same physical machine



► Disadvantages

- Less efficient than a real machine when it accesses the hardware indirectly, but feasible with modern computer

► Examples

- IBM VM/370 (1970s technology!)
- VMWare (https://de.wikipedia.org/wiki/VMware_Workstation)
- Microsoft Virtual PC / Hyper-V
- And many more ...



► VMs on a single host

- Share same OS
- Have the same OS and ISA
- Have different OS but same ISA
 - VMs run their own operating system (guest operating system) $\Rightarrow$ multiple OS environments can co-exist on the same computer
- Have different OS and ISA
 - Ability to provide an ISA that is somewhat different from that of the real machine

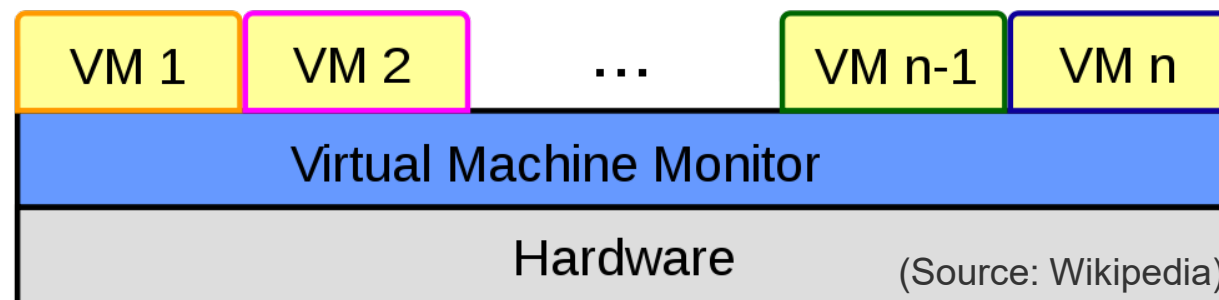


- ▶ Hypervisor-based virtualization
 - Software virtualization
 - Hardware-supported virtualization
 - Paravirtualization
 - Mostly a combination of different virtualization techniques used in practice
- ▶ OS level virtualization



Hypervisor-Based Virtualization

- ▶ Hypervisor also called Virtual Machine Monitor (VMM)
- ▶ Software that provides resources to VMs and handles real I/O devices
 - VMs possibly see real hardware: CPUs, RAM, I/O devices, ...
 - Emulates generic virtual I/O devices for guest
 - Details depend on specific virtualization technique
- ▶ Monitors VMs
- ▶ Enforces
 - Access limitations
 - Resource limitations
- ▶ Guest OS may be different from host OS
- ▶ Guest code runs on native machine in user mode
- ▶ Traps to VMM on
 - Privileged instructions
 - Access to protected resources





► Software virtualization

- Does not require hardware support
- Traditional approach
- Slow $\Rightarrow$ usage decreases
- Useful for cross virtualization
 - E.g.: ARM processor virtualized on x86 machine
 - Used for development of smartphone applications on desktops
- Tools: VMware, QEMU (Quick Emulator), VirtualBox

► Hardware-supported virtualization

- Requires hardware support
 - x86 used to be not-virtualizable in contrast to, e.g., PowerPC
- Since about 2007: hardware support for virtualization on x86 through extensions
- Hardly any performance loss compared to physical machine
- Tools: Hyper-V, VirtualBox, kvm with QEMU



► Paravirtualization

- VM knows about virtualization
- Better performance than software virtualization
- Example
 - VM knows NIC is virtualized
 - Uses different driver that performs operations in memory instead of virtualized network stack
 - Improved performance
- Tools: Xen, QEMU with virtIO devices, VirtualBox

► Mostly a combination of different virtualization techniques used in practice

- Example
 - Hardware-supported virtualization for CPU, RAM
 - Paravirtualization for NIC
- Advantages
 - More flexibility
 - Not all physical machines support all kinds of hardware virtualization
 - Facilitates VM migration
 - VM migration with hardware-supported NIC virtualization requires same device and same (free) bus ID for device on target host



- ▶ VT-x: Enables basic hardware acceleration
- ▶ VT-d: IOMMU (I/O memory management unit)
 - Translates virtual device addresses to physical device addresses
 - Similar to MMU
 - Translates between virtual and physical memory addresses
 - Needed to pass through physical hardware from host to VM (e.g., NIC)
- ▶ VT-c
 - Virtual machine device queues (VMDq)
 - Enables multiple I/O queues per NIC
- Facilitates packet forwarding between host and VM in hardware
 - Saves memory copies
- Aside: PCI Express (PCIe) is standard for high-speed communication within physical machine
- Single root I/O virtualization (SR-IOV)
 - Extension of PCIe
 - Physical function (PF)
Normal PCI device
 - Virtual function (VF)
Lightweight PCI device
Requires PF to work



► Hyper-V

- Microsoft
- Hypervisor for x86
- Hardware-accelerated virtualization on Windows servers
- Typically used in Windows-based datacenters

► VMware

- VMware
- Hypervisor for x86
- Host runs on Linux, Windows, (Mac) OS X
- Software virtualization

► VirtualBox

- Oracle
- Hypervisor for x86
- Host runs on Linux, Windows, Solaris, OS X
- Supports
 - Software virtualization
 - Paravirtualization
 - Hardware-supported virtualization



Hypervisor-Based Virtualization Solutions (2)

► Xen

- Hypervisor for x86
- Citrix Systems
- Griechisch ξένος, *xénos*: Fremder
- Provides OS for physical machine
 - Privileged management domain
 - Unprivileged VMs
- Often used for paravirtualization

► QEMU (Quick Emulator)

- Linux tool
- General purpose hypervisor
- Supports cross-virtualization

► kvm (kernel virtual machine monitor)

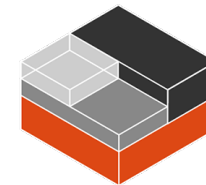
- Part of Linux kernel
- Hypervisor providing hardware support
- Not standalone, requires, e.g., QEMU or VirtualBox

► Typical virtualization for Linux on x86

- Combine kvm and QEMU
- Manage VMs with libvirt



- ▶ Objective: lightweight isolation of parallel Linux systems and applications with regard to resources and security
- ▶ Does not use a hypervisor
- ▶ VMs called „containers“
- ▶ Mechanisms
 - Containers share kernel of host system
 - Linux control groups (cgroups)
 - Control for hardware access
 - Resource limitation
 - Process prioritization
 - Isolated namespaces
 - Enable own process and network space per container on host
- ▶ Tools: LXC, Docker
- ▶ Orchestrierung von Containern: Kubernetes („Steuerermann“)



LXC





- ▶ As Docker became more popular, it was desirable to run it across different OSs
 - E.g., development on a local Windows machine
- ▶ First versions of Docker for Windows used Hyper-V to spin up a Linux VM to run the Docker containers
 - Not resource-friendly
- ▶ As of Docker for Windows version 2.3.0.2 (11.05.2020), it uses the WSL2 backend
 - **W**indows **S**ubsystem for **L**inux
 - Windows-native way to run Linux executables
 - Does not require a VM
 - Uses less resources to run the containers



Related: How to run Windows apps on Linux and vice-versa?

► Wine (Wine Is Not an Emulator)

- [https://en.wikipedia.org/wiki/Wine_\(software\)](https://en.wikipedia.org/wiki/Wine_(software))
- Free and open-source compatibility layer to allow application software and computer games developed for Microsoft Windows to run on Unix-like operating systems.
- Developers can compile Windows applications against WineLib to help port them to Unix-like systems.
- No code emulation or virtualization occurs.
- The goal of Wine is to implement the Windows APIs fully or partially that are required by programs that the users of Wine wish to run on top of a Unix-like system.
- Wine is primarily developed for Linux and macOS.



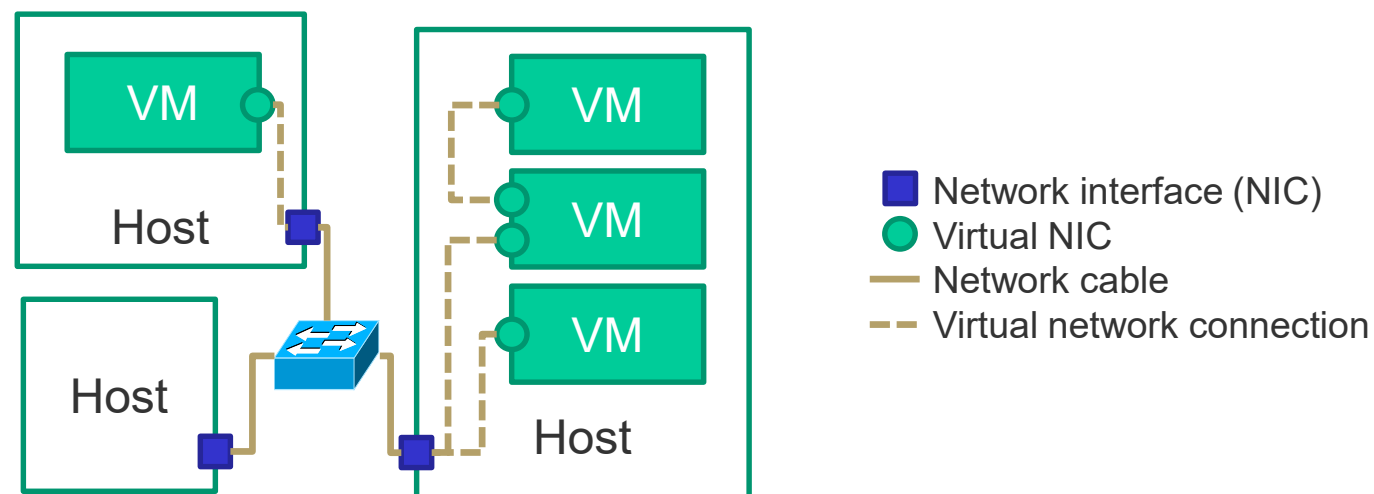
► Windows Subsystem for Linux (WSL)

- https://en.wikipedia.org/wiki/Windows_Subsystem_for_Linux
- Feature of Microsoft Windows
 - Allows developers to run a Linux environment without the need for a separate virtual machine or dual booting.
- WSL 1, since 8/2016
 - Acts as a compatibility layer for running Linux binary executables by implementing Linux system calls on the Windows kernel.
- WSL 2, since 5/2019
 - Runs inside a managed virtual machine that implements the full Linux kernel.
 - WSL 2 is compatible with more Linux binaries than WSL 1, as not all system calls were implemented in WSL 1.





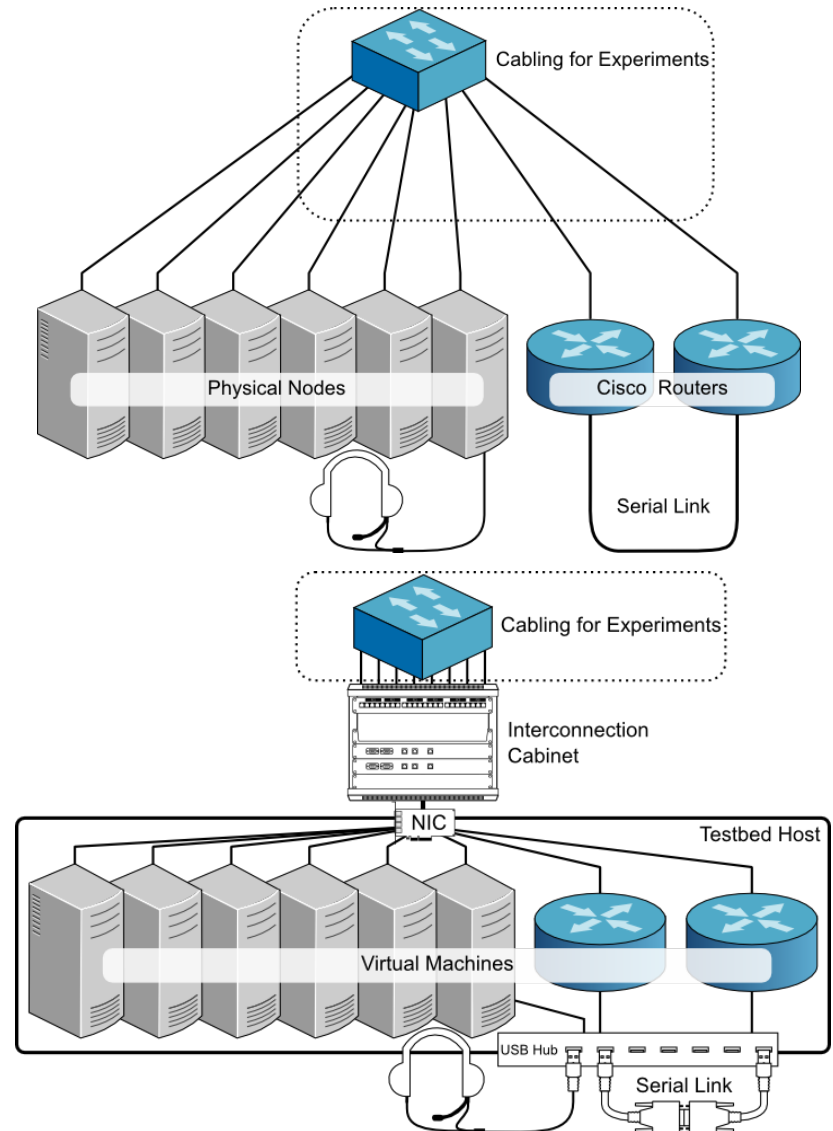
- ▶ Hosts communicate via network interface cards (NICs) over a network
- ▶ VMs communicate via virtual NICs
 - Within a host
 - With other hosts and VMs over a network
 - **Physical NICs need to be shared**





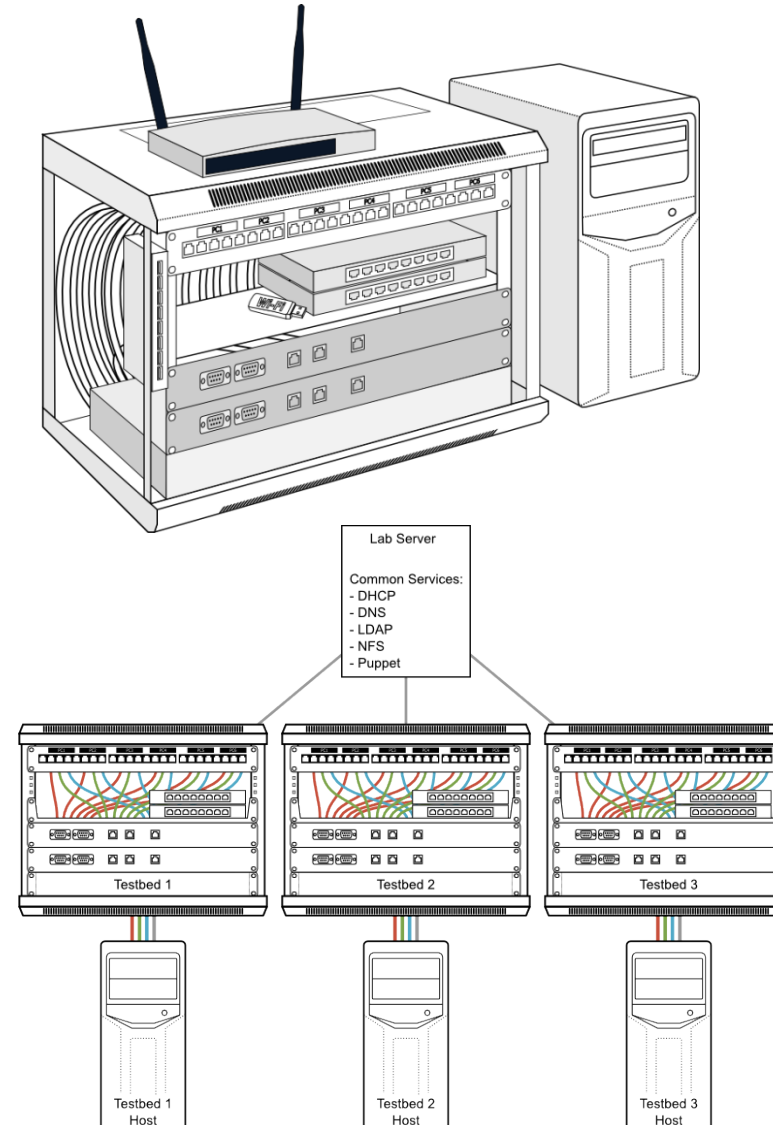
A Virtualized Testbed with Physical Outlets for Hands-on Computer Networking Education

- ▶ Practical networking courses are important
- ▶ Physical testbeds
 - Real hands-on experience
 - Drawbacks
 - Space
 - Heat
 - Energy consumption
 - Expensive
- ▶ Virtual testbeds
 - Solved drawbacks of physical testbeds
 - Drawbacks
 - No real hands-on experience
- ▶ Hands-on virtual testbeds
 - Solved drawbacks of physical testbeds
 - Preserves real hands-on experience
 - Challenging





- ▶ Testbed virtualization platform
 - Hardware support for virtualization
 - Ubuntu 14.04
 - Hypervisor and virtualization framework
- ▶ Hardware
 - CPU: Intel Xeon E3
 - NIC: I350-T4
- ▶ Lab infrastructure
 - Centralized accounts
 - Centralized home directories
 - Network services: DNS, DHCP
 - Configuration management





Providing Physical Access to Virtual Ethernet Interfaces

► Each of the 6 VMs

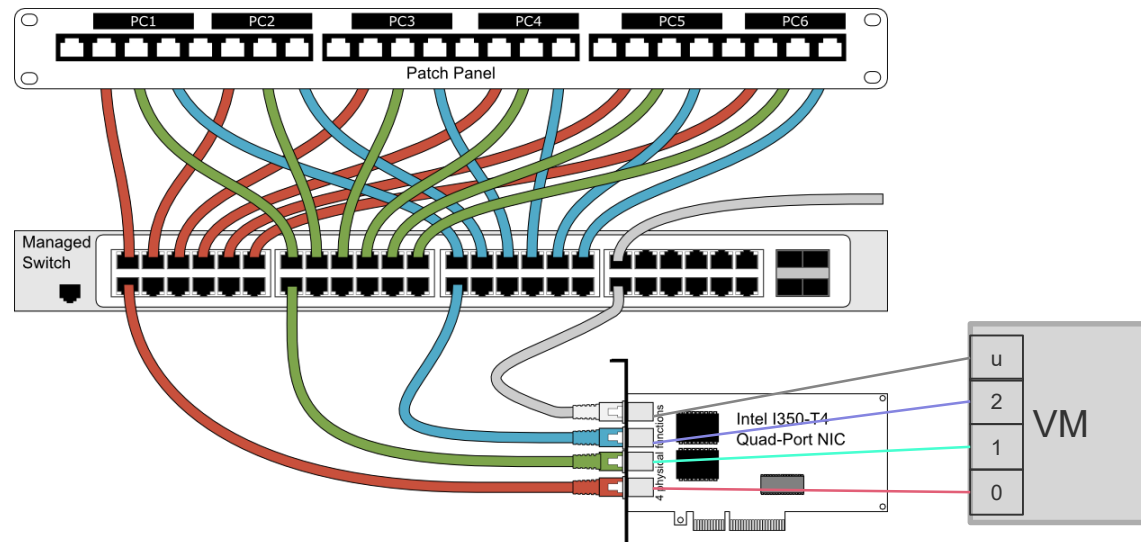
- 3 Ethernet interfaces for experiments (0..2)
- 1 Ethernet interface for Lab infrastructure (u)
- → 24 virtual Ethernet interfaces and physical Ethernet ports required

► Virtual interfaces

- No dedicated NICs possible
 - Limited number of PCI ports
- Multiple virtual interfaces per physical interface
- **Multiplexing required**

► Physical outlets

- Dedicated outlets possible
 - Patch panel
- **Demultiplexing required**



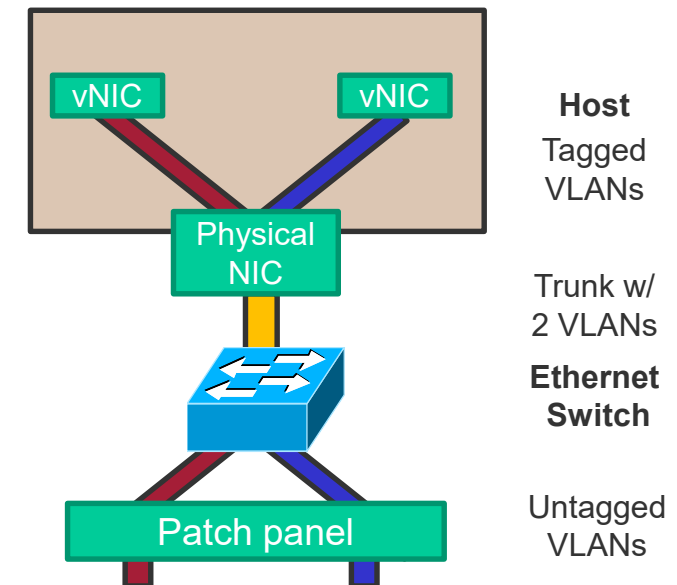
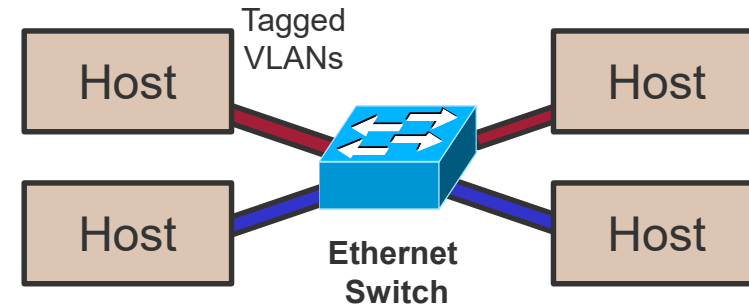
M. Schmidt et al.: "A Virtualized Testbed with Physical Outlets for Hands-on Computer Networking Education", ACM SIGITE 2014

M. Schmidt et al.: A Semi-Virtualized Testbed Cluster with a Centralized Server for Networking Education, Intl. Teletraffic Congress (ITC), September 2018



Virtual LAN (VLAN)

- ▶ Identifier (VLAN tag) within an Ethernet frame
- ▶ Streams belong to different logical LANs
 - Example: one for employees, one for students, one for printers
 - Strong separation: only hosts in same VLAN can communicate with each other
- ▶ VLAN tags may be used for multiplexing/demultiplexing traffic over a single NIC





Providing Physical Access to Virtual Ethernet Interfaces

- ▶ Access to interfaces of 6 VMs and 2 virtual routers required
- ▶ 4-port network interface card
 - Provides 4 PFs
 - Each PF provides 7 VFs
 - VFs passed-through to VMs
 - PF acts as bridge in VEPA mode
- ▶ (De-)multiplexing of interfaces
 - Dedicated VLAN per virtual interface
 - Multiple VLANs per cable between PF and managed switch (trunk)
 - VLAN-tags stripped/added at PF and managed switch

