SPECIALISATION: CLUSTERING

Server Virtualization Systems
Dual-Hypervisor Clustering

BTS Cloud Computing • Module VIRCL



VMWARE vSPHERE

vs



PROXMOX VE

Presented by **Andrea Girotto** & **Guilherme Videira Marques** (Team 4)

ROADMAP

Presentation Agenda

01

The Problem

Clustering basics, quorum voting logic, and the split-brain risk.

02

Guarantees

Scheduled live migration vs. unplanned HA failover.

03

Architecture

Our nested 3-node topology and the ESXi & Proxmox setup.

04

Under the Hood

How multi-master really works: Corosync's Totem ring, pmxcfs & fsync.

05

Compare, Demo, Rules

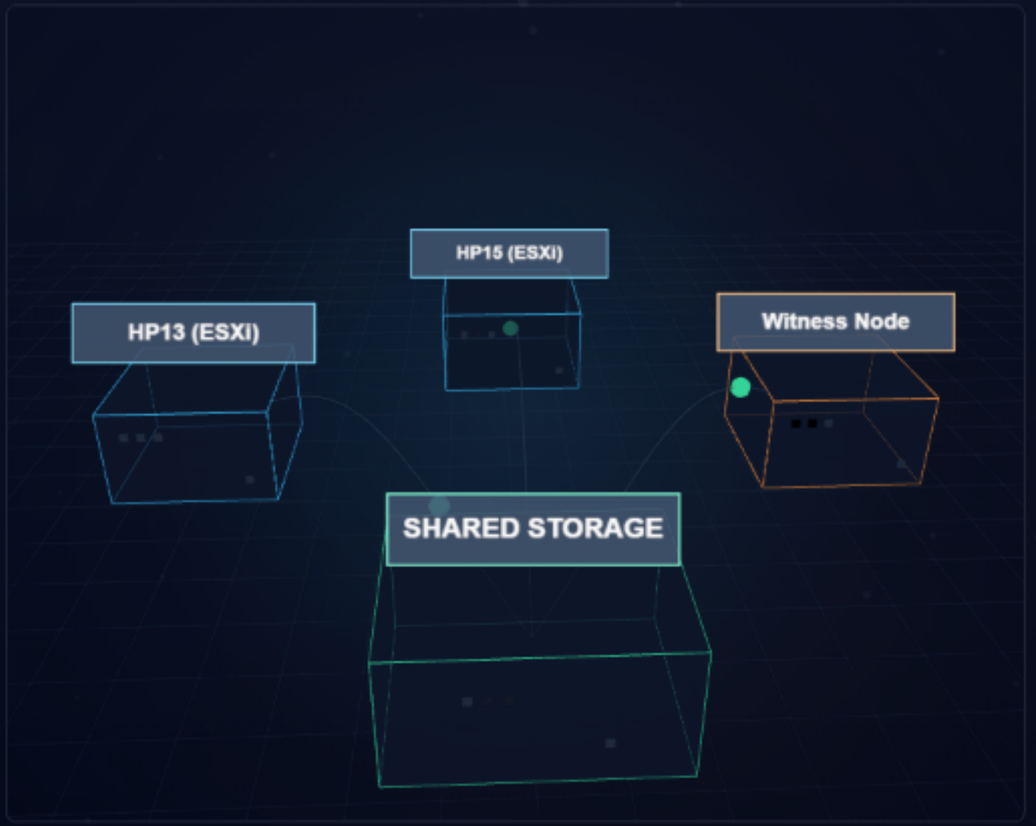
Side-by-side comparison, live migration, lab rules, and final takeaway.

FUNDAMENTALS

What is a Hypervisor Cluster?

Concept: Several physical hosts behave like one platform, while each VM still runs on one host at a time.

- **One control plane:** Manage hosts and VMs from a shared inventory
- **Shared VM disks:** Eligible hosts can access the same datastore
- **Live migration:** Move a running VM before maintenance
- **HA recovery:** Restart a VM elsewhere after a host failure
- **Consensus:** Quorum decides who is allowed to write shared state



MATHEMATICAL CONSENSUS

The Core Problem: Quorum & Split-Brain

2-Node Cluster (Unsafe)

With only two voters, a host failure or network split leaves no tie-breaker. One remaining vote cannot prove safe ownership of shared state.

Required:	votes ≥ ⌊n/2⌋ + 1
Nodes (n):	2
Quorum Target:	2 votes
Votes (1 Failed):	1 vote



NO MAJORITY / SPLIT-BRAIN RISK

3-Node Cluster (Safe)

A third vote creates a majority. If one node disappears, the two remaining voters can continue and reject the isolated side.

Required:	votes ≥ ⌊n/2⌋ + 1
Nodes (n):	3
Quorum Target:	2 votes
Votes (1 Failed):	2 votes



QUORUM SECURED

OPERATIONS

Planned Migration vs. Unplanned Failover

Live Migration

(vMotion / Online)

- **Trigger:** Planned maintenance or load balancing
- **Execution:** RAM is copied while the VM keeps running
- **User impact:** Brief stun, usually not visible
- **State:** Active memory and sessions are preserved

HA Failover

(High Availability)

- **Trigger:** Host crash, power loss, or heartbeat timeout
- **Execution:** Cluster starts the VM on another host
- **User impact:** Reboot downtime
- **State:** RAM is lost; the guest OS boots cleanly

DESIGN

Architecture: Nested 3-Node Topology

PROBLEM

Each platform only had two physical hosts. With two voters, losing one side means losing quorum.

DESIGN MOVE

The third voter is a **nested VM** running inside the opposite hypervisor platform.

ESXI QUORUM

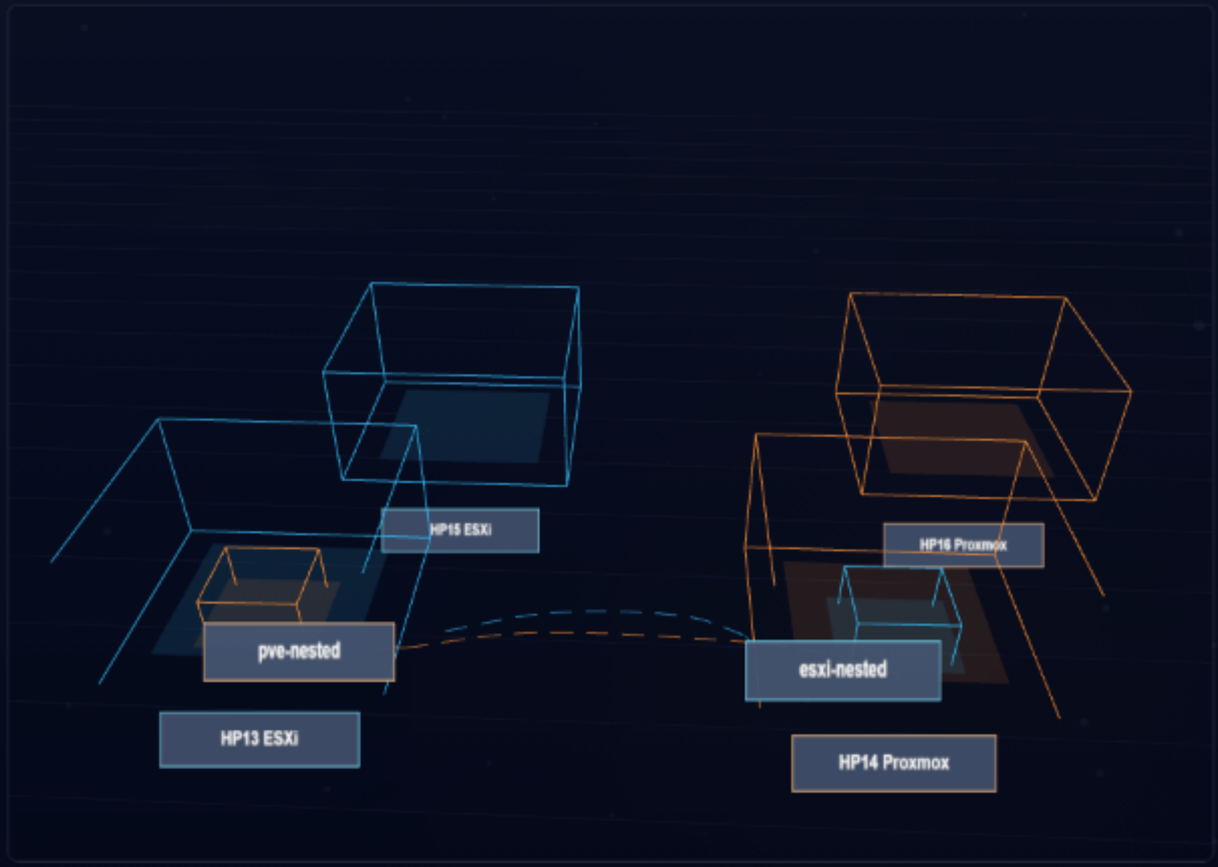
HP13 + HP15 + esxi-nested

esxi-nested runs inside HP14 Proxmox

PROXMOX QUORUM

HP14 + HP16 + pve-nested

pve-nested runs inside HP13 ESXi



HYPERVISOR 1

VMware ESXi + vCenter Setup

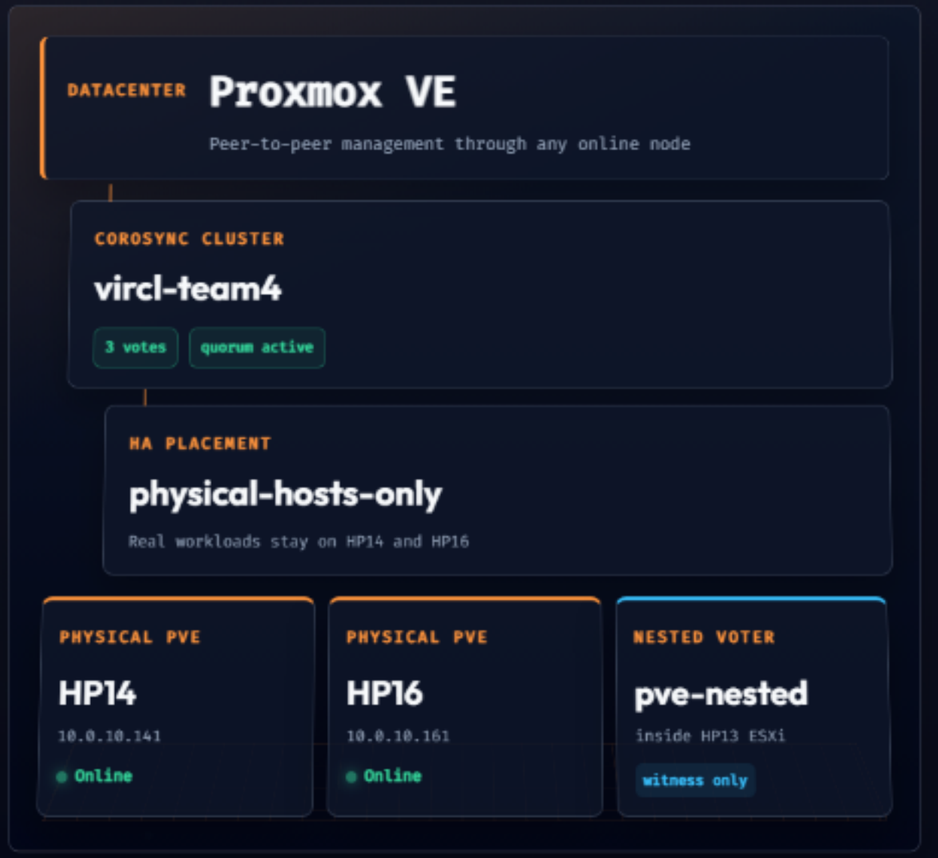
- **Management Layer:** VCSA provides one control point for the ESXi environment
- **Cluster Inventory:** VIRCL-Cluster contains HP13, HP15, and esxi-nested
- **Nested ESXi Host:** Runs inside HP14 Proxmox to complete the lab topology
- **Automation:** vSphere HA enabled, DRS set to Fully Automated



HYPERVISOR 2

Proxmox VE + Corosync Setup

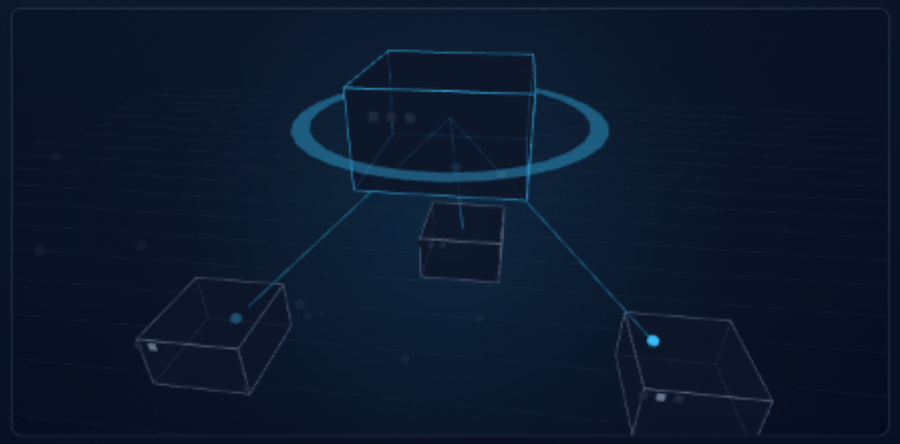
- **Management Model:** Peer-to-peer access through any online node
- **Cluster State:** vircl-team4 uses Corosync for membership and quorum
- **Voting Members:** HP14, HP16, and pve-nested provide three votes
- **HA Placement:** Real workloads stay on HP14 and HP16 through physical-hosts-only



UNDER THE HOOD

The Multi-Master Trade-off

Centralized Master vCenter / FDM



A central service keeps the cluster view consistent and tells hosts what to do. It is easier to reason about, but that control layer must be protected because it can become a **single point of failure**.

Multi-Master Mesh Proxmox



Each node can manage the cluster, start migrations, and take part in HA decisions. That removes the obvious **single point of failure**, but the nodes must agree on every change.

UNDER THE HOOD • CONSENSUS

Corosync & the Totem Token Ring

Corosync keeps the cluster in one shared order: every node sees the same events, in the same sequence.

- One **token** moves around a logical ring.
- Only the node holding the token may send cluster updates.
- After sending, it passes the token to the next node.
- When the token returns, the sender knows the update reached the cluster.
- Result: independent nodes behave like one ordered system.



UNDER THE HOOD • STATE SYNC

pmxcfs: Config Sync Down to the Disk

pmxcfs looks like normal files, but every edit becomes replicated cluster state.

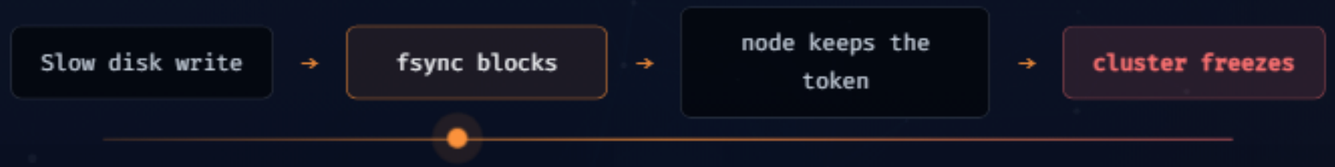
- **FUSE** exposes it as `/etc/pve`, while pmxcfs keeps the real state in a small in-memory SQLite DB.
- A config change waits for the **Totem token**, then gets multicast in the same order to every node.
- Each node writes the update locally and must finish **fsync** before the change is truly committed.
- That is why cluster consistency is tied directly to **physical disk speed**.



UNDER THE HOOD • CONSEQUENCES

The Hardware Reality

In Proxmox HA, consistency is limited by the slowest physical write.



Dedicated Corosync network < 5 ms

Keep cluster traffic on a low-latency link. Jitter delays token handoff and can look like node failure.

Enterprise SSDs with power-loss protection

Drives with capacitors can safely acknowledge fsync quickly, instead of stalling the cluster on flash writes.

⚠ **Consumer-grade SSDs** can pause on fsync → Corosync sees a stalled node → healthy nodes may **self-fence** and leave the cluster.

COMPARISON

ESXi vs. Proxmox: Side by Side

Capability	VMware vSphere (ESXi)	Proxmox VE
Management	Centralized vCenter database and services.	Peer-to-peer management from any node.
Cluster Engine	vSphere HA / FDM with controller-style coordination.	Corosync consensus plus pmxcfs state sync.
Load Balancing	DRS can automate placement and balancing.	HA groups and constraints guide placement.
Quorum Consensus	Witness support for 2-node or stretched designs.	Strict majority quorum; QDevice adds witness votes.

TAKEAWAY **vSphere favors controller-led automation; Proxmox favors peer consensus and transparent control.**

LIVE DEMO

Live Demo: Online VM Migration

RELIABLE DEMO PATH

```
# VM 101: Lubuntu, 40 GB disk on shared NAS

hp14# qm migrate 101 hp16 --online
```




```
• Pre-check: HP14 + HP16 online... OK
• VM disk already on shared NAS storage... OK
• Guest stays reachable during migration... PING
• VM state transfers while workload stays
online... OK
• Final ownership switches to HP16... OK
```

What the demo proves

01 Shared storage removes the disk problem
The VM disk already lives on NAS, so the host handoff only moves runtime state.

02 The workload stays online
Lubuntu keeps running while Proxmox transfers the active VM state.

03 The cluster coordinates ownership
Corosync and pmxcfs keep the VM definition consistent during the handoff.

04 Control moves from HP14 to HP16
The visible result is the same VM now owned by the target physical host.

LAB INSIGHTS

Lab Rules That Actually Matter

Design for quorum

OK

Use an odd number of votes

Three is the minimum; odd vote counts avoid deadlocks and keep a clear majority.

OK

Use nested nodes only as witnesses

They are useful for quorum, but real VMs should stay on HP14 and HP16.

OK

Allow nested Proxmox traffic

On ESXi, Forged Transmits must be accepted or nested MAC traffic can disappear.

Avoid false promises

NO

Do not trust a 2-node cluster

One failure is enough to lose quorum, and split-brain becomes a real risk.

NO

Do not treat shared storage as optional

Without shared storage, migration and HA restart targets cannot safely access the same VM disks.

NO

Do not oversell HA failover

HA restarts a crashed VM on another host; it does not keep the VM running.

FINAL TAKEAWAY

Clustering is not a product feature. It is a discipline.

Shared storage
Every surviving host must reach the VM disks.

Quorum math
The cluster must always know who owns the truth.

Clear limits
Migration, HA, and failover are different promises.

Questions & Answers

VIRCL Team 4 • Andrea Girotto & Guilherme Videira Marques