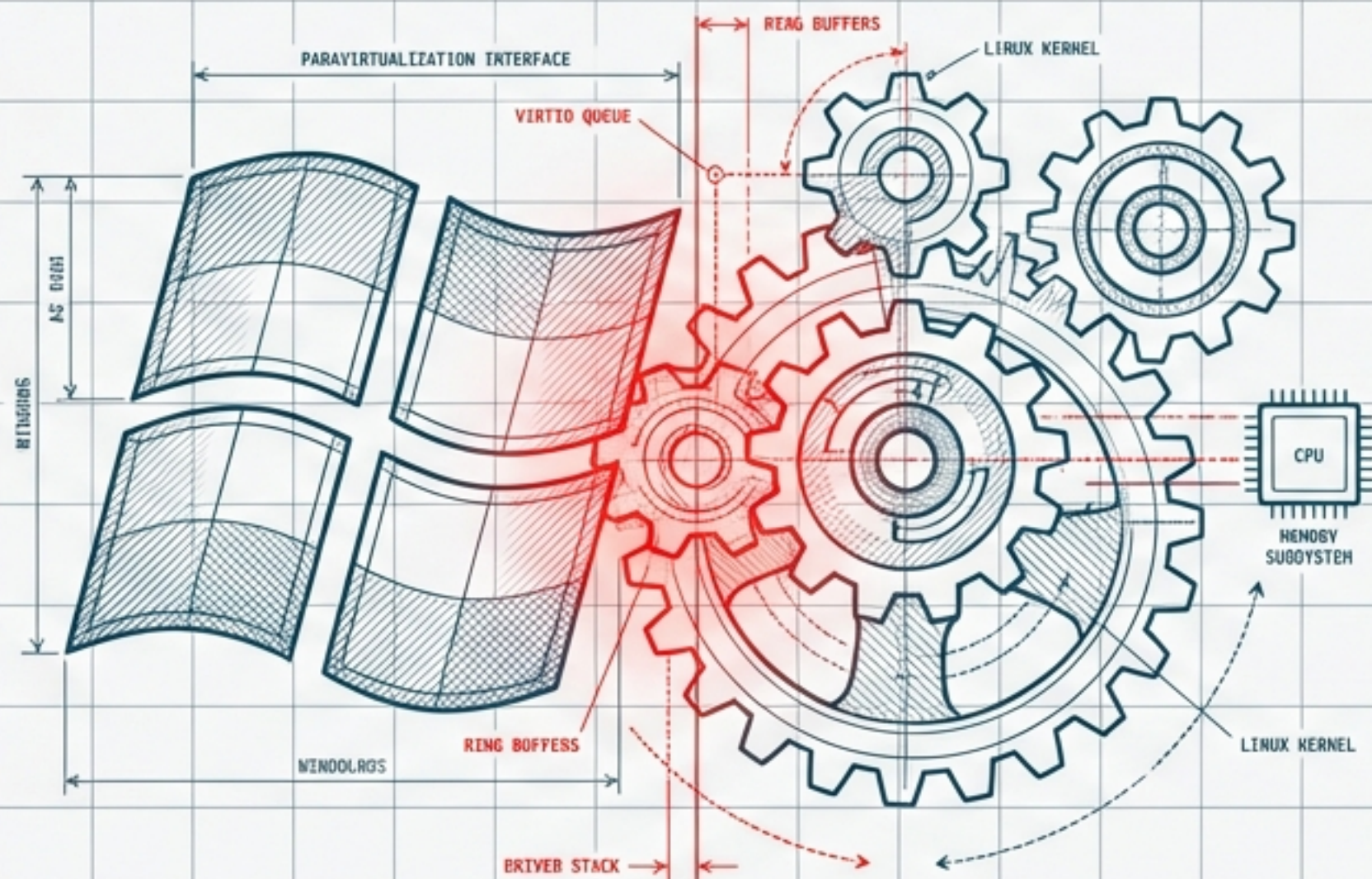


# Virtio-Win 项目深度解析：在 KVM 环境中释放 Windows 性能

从半虚拟化架构到企业级 WHQL 认证的完整工程视图



[github.com/virtio-win/kvm-guest-drivers-windows](https://github.com/virtio-win/kvm-guest-drivers-windows)



# 项目概览：连接 Windows 与 KVM 的原生性能桥梁

**Virtio-Win** 是一套专为 Windows 客户机设计的半虚拟化 (Paravirtualized) 驱动程序，旨在绕过传统硬件模拟的性能瓶颈。



## 性能 (Performance)

通过共享内存机制 (Virtqueues) 替代高开销的上下文切换。



## 治理 (Stewardship)

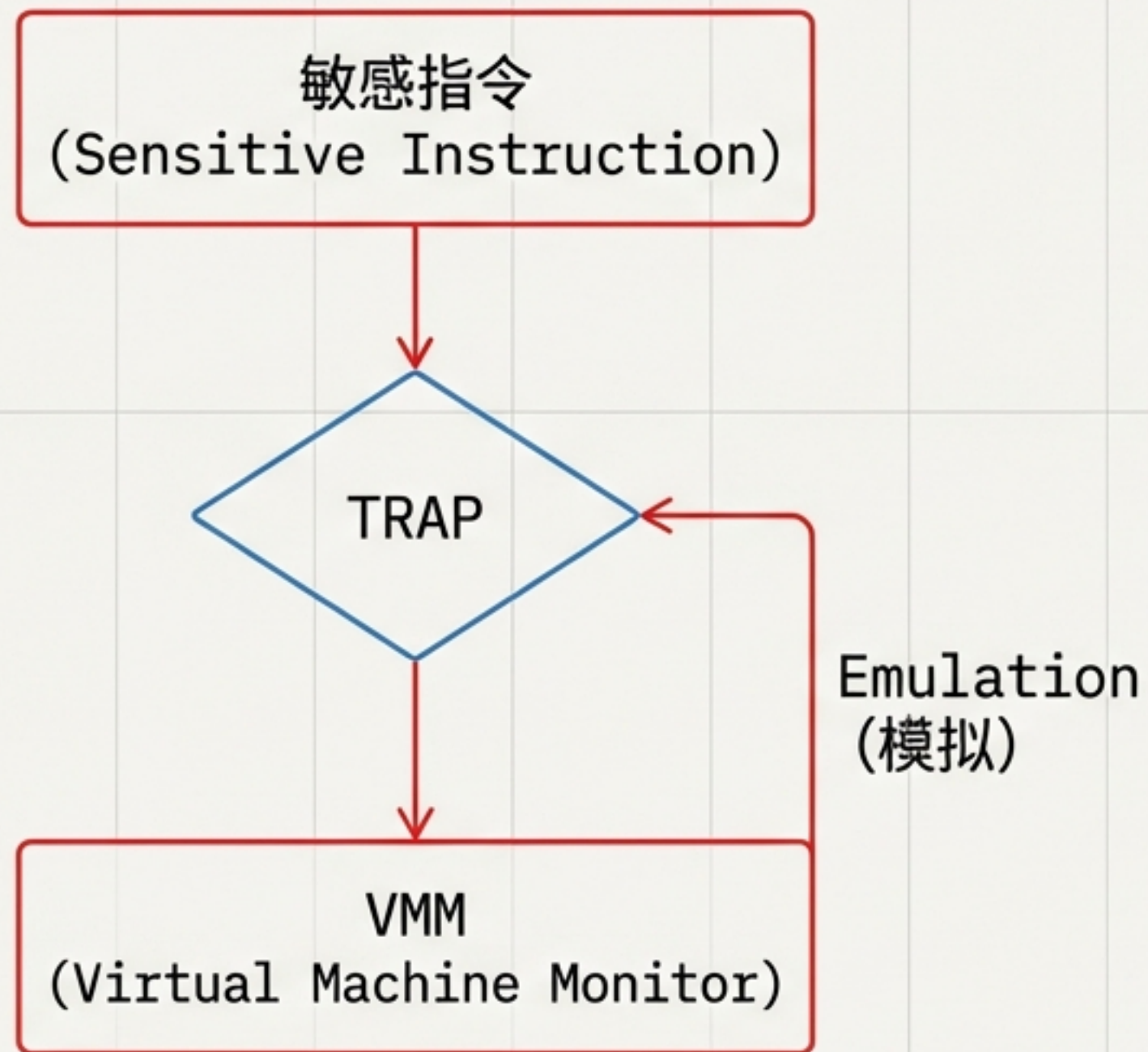
由 **Red Hat** 赞助, Daynix Computing (Yan Vugenfirer) 维护, 拥有活跃的开源社区。



## 认证 (Certification)

提供从开源构建 (**GitHub**) 到通过微软 WHQL 认证 (**Red Hat Enterprise Linux**) 的多种交付形态。

# 性能瓶颈：传统模拟与上下文切换开销

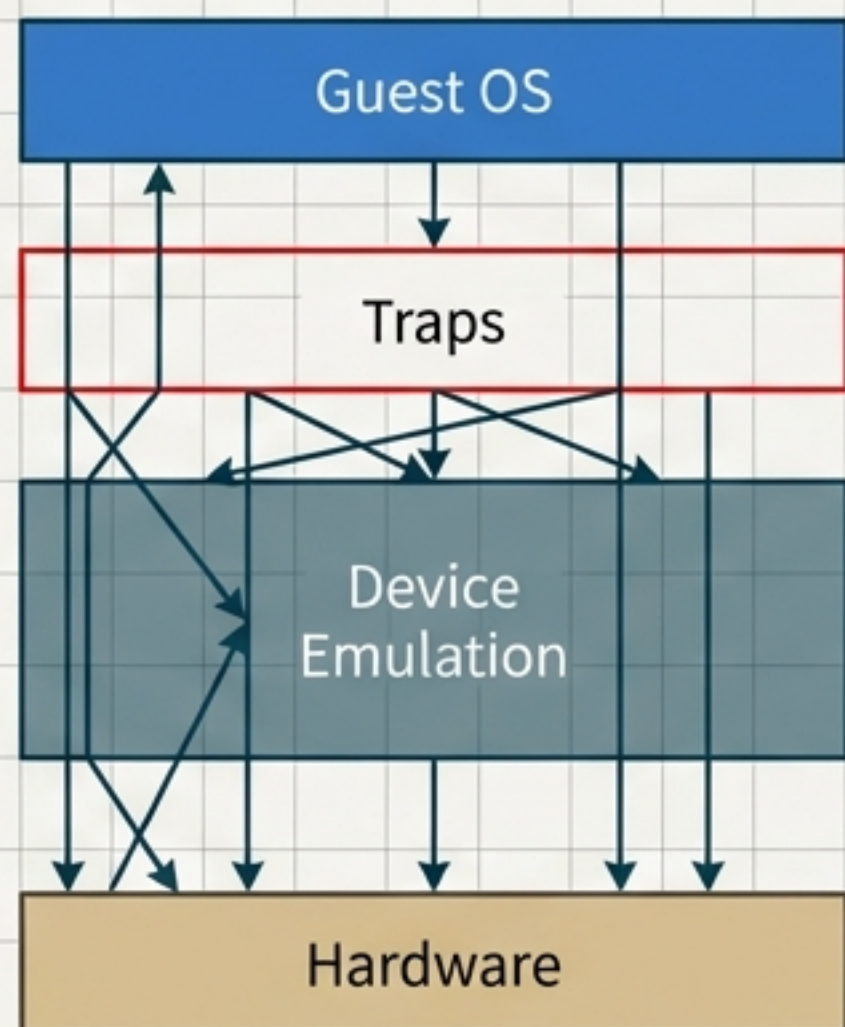


- 在没有 VirtIO 之前，KVM 必须模拟物理硬件（如 RTL8139 网卡或 IDE 控制器）。
- 每次 I/O 操作都会触发 VMExit，导致 CPU 在 Ring 0 (Guest) 和 Root Mode (Host) 之间频繁切换。
- 影响 (Impact)：高延迟与低吞吐量，CPU 周期浪费在模拟逻辑上而非业务处理上。

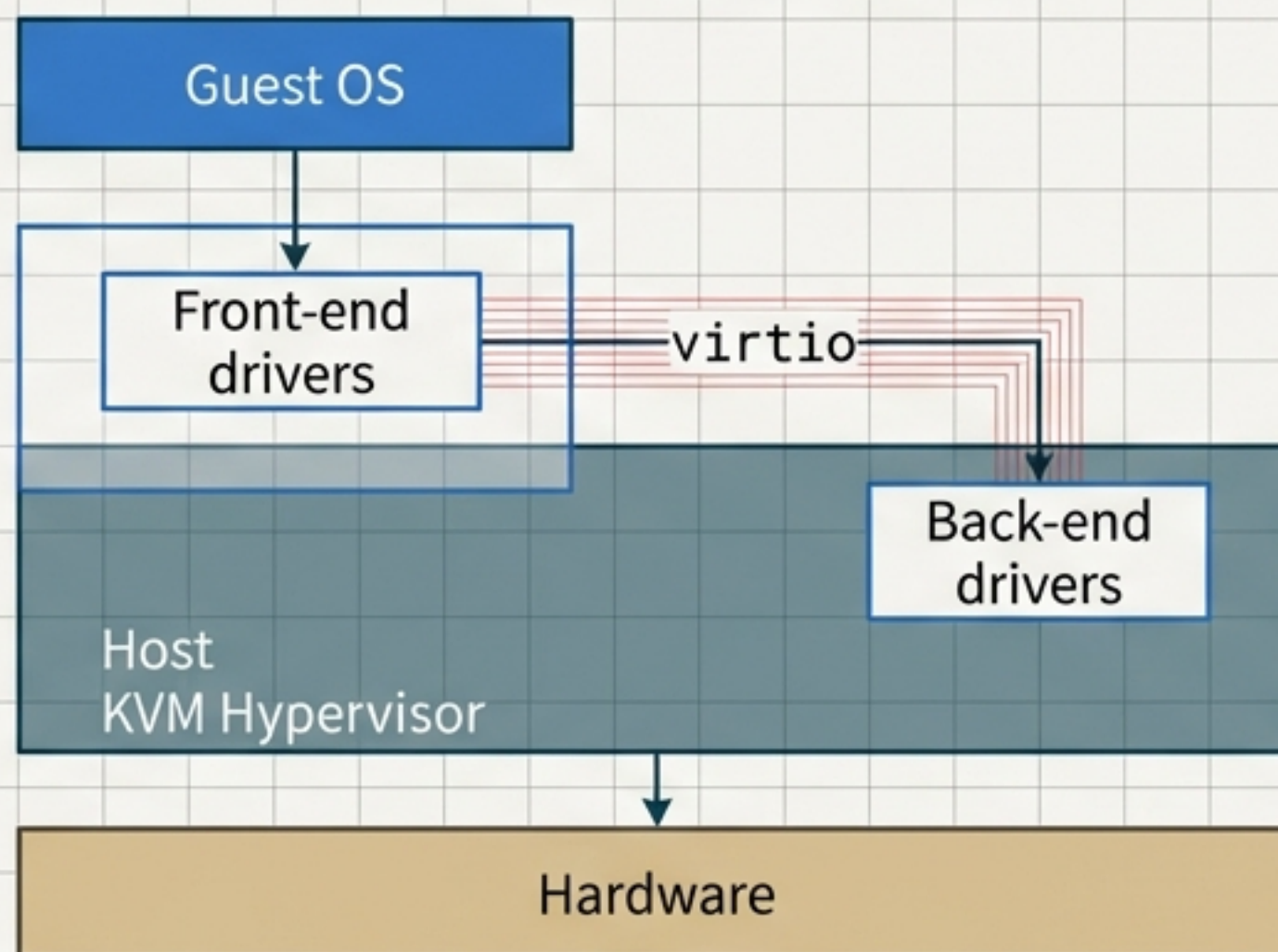
Source: GitHub Repository - KVM/QEMU Windows guest drivers

# 架构变革：VirtIO 标准与半虚拟化

## Legacy Emulation (传统模拟)



## VirtIO (Paravirtualization)



**感知 (Awareness):** Guest OS “知道”自己运行在虚拟环境中。

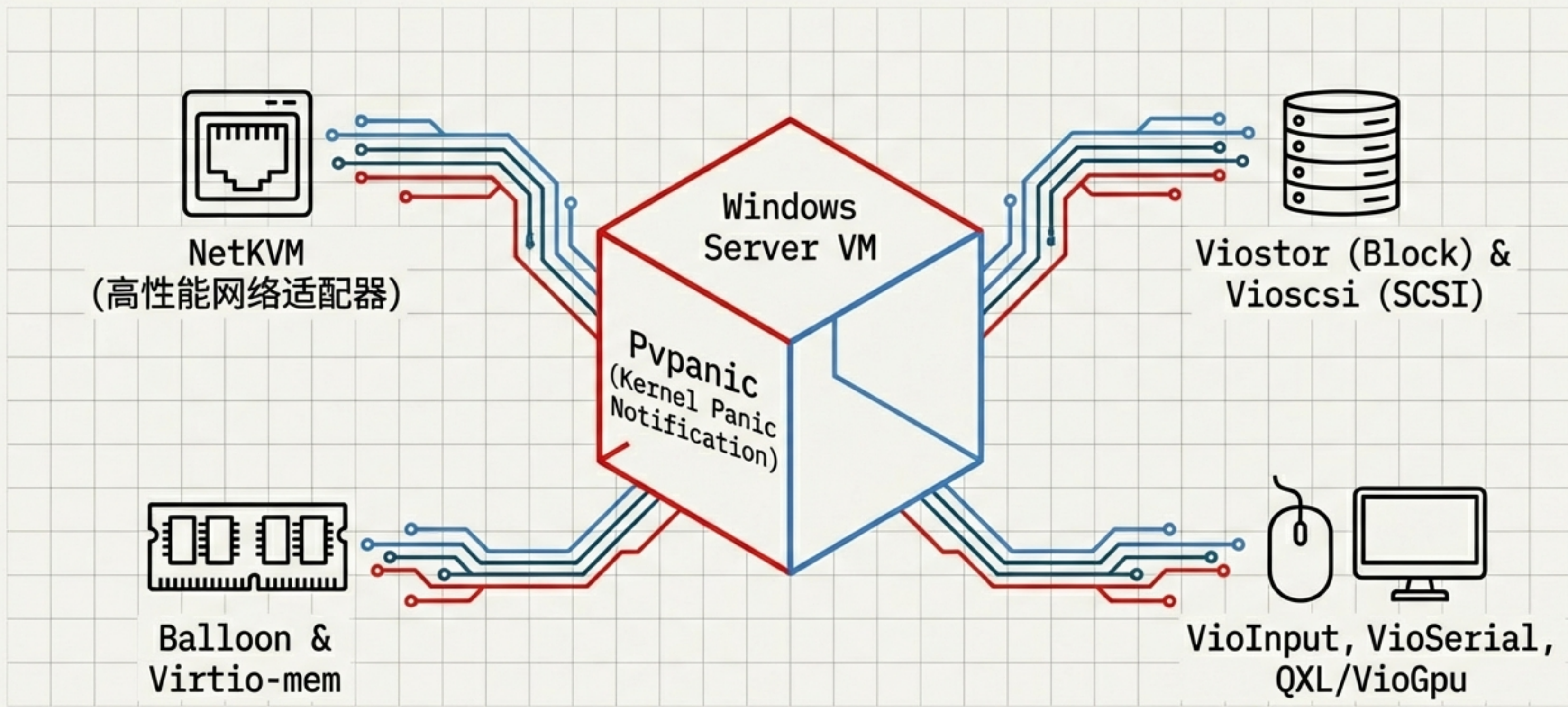


**Virtqueues:** 使用共享内存环进行批量数据传输，无模拟开销。



**结果:** 消除不必要的 Trap 操作，Guest 直接通过标准接口协作。

# 驱动程序生态系统全景

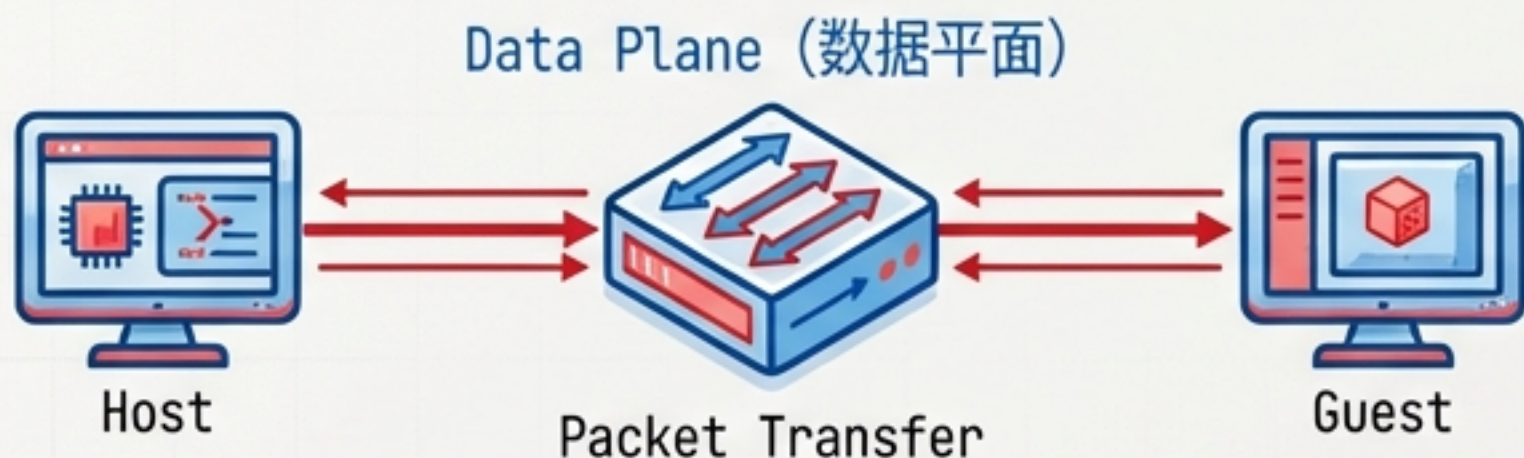
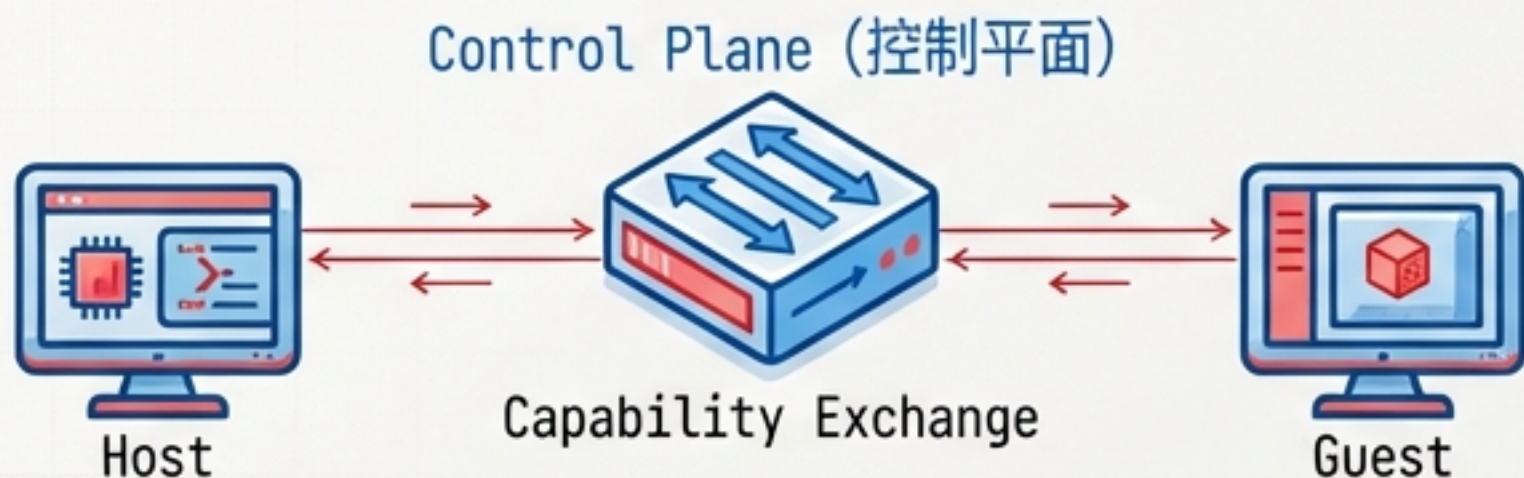


Source: GitHub Repository - KVM/QEMU Windows guest drivers

# 网络性能深度解析：NetKVM

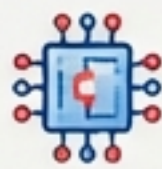



## Control vs Data Plane

JetBrains Mono










## Technical Specifications List

JetBrains Mono

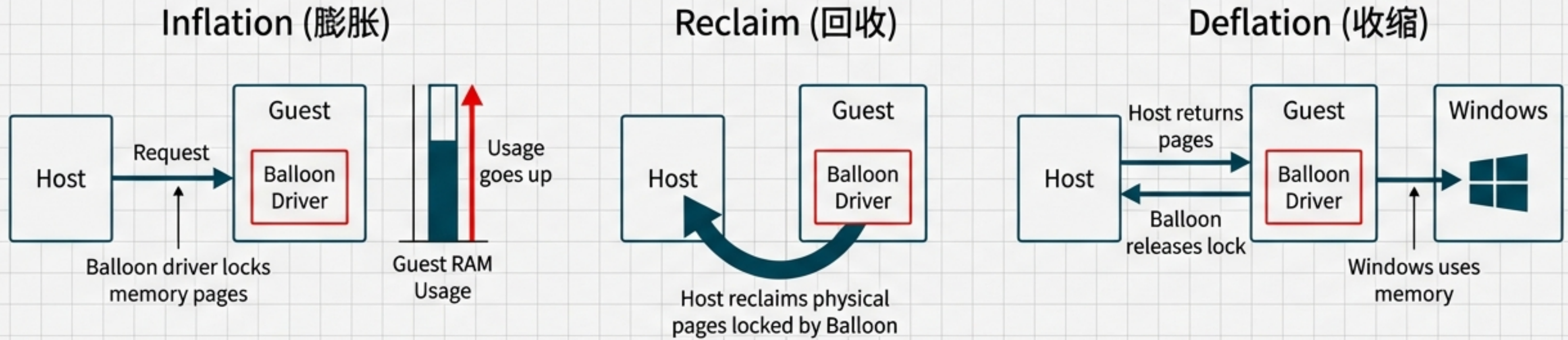
-  Architecture: 基于 NDIS Miniport 驱动模型，绕过传统网络堆栈开销。
-  RSS (Receive Side Scaling): 利用 Host 端的 eBPF 程序进行流量控制，分散 vCPU 负载。
-  Offloading: 支持 Checksum, LSO, RSC。
-  Stability: 解决高并发下的缓冲区短缺 (GitHub Issue #1058) 与 Spinlock 优化。

# 存储演进：从 Viostor 到 Vioscsi

Viostor (Legacy)	Vioscsi (Modern Standard)
<ul style="list-style-type: none"><li> 类型：基础块设备驱动</li><li> 局限：功能简单，缺乏高级命令支持</li><li> 适用：旧版兼容性场景</li></ul>	<ul style="list-style-type: none"><li> 类型：完整的 SCSI HBA 驱动</li><li> 并发 (Concurrency)：单控制器支持数百个磁盘</li><li> 特性：支持 TRIM/UNMAP (精简置备回收)</li><li> 集成：完美适配 Ceph RBD 和 NVMe 后端</li></ul>

Source code directory: `/vioscsi` vs `/viostor`

# 动态资源管理：Balloon 与 Virtio-Mem



**智能机制：**相比 Host 强制 Swap，Balloon 允许 Windows 内存管理器决定通过减少文件缓存来释放内存，最小化性能损耗。

**新技术：**virtio-mem 支持物理内存的热插拔 (Hot-plug/Unplug)。

# 辅助组件：图形加速与客户机代理



## QEMU Guest Agent (QGA)

- **VSS Integration:** 通过 Volume Shadow Copy Service 实现文件系统冻结，确保数据一致性备份。
- **Management:** 允许 Host 执行 Guest 内命令（如重置密码、查询 IP）。



## Graphics: VioGpu / QXLDDOD

- 替代传统 VGA 模拟，提供 2D/3D 加速。
- 支持多显示器配置和鼠标无缝集成 (Mouse Integration)。

Noto Sans SC

# 工程流水线：CI/CD 与自动化测试



Noto Sans SC

AutoHCK: Daynix 开发的自动化框架，自动运行微软 HCK/HLK 测试套件，确保驱动稳定性。

# 认证壁垒：WHQL 与数字签名

## 挑战：Windows 10/11 (Secure Boot) 拒绝加载未签名驱动。

现代 Windows 系统在启用了安全启动 (Secure Boot) 的环境下，出于安全考虑，会强制要求所有加载的驱动程序必须经过微软的数字签名认证。未通过认证的驱动将被阻止运行，这成为开源驱动项目面临的主要障碍。



# 项目治理与维护者

## Daynix Computing

核心维护者 (Maintainer)  
& 知识中心

负责 upstream 开发、HCK  
自动化及 Issue 修复。

Yan Vugenfirer (CEO)

## Red Hat

企业级赞助商

集成代码至 RHEL 生态，提  
供商业支持。

## Fedora Community

上游分发渠道

提供最新的社区版构建。

# 最佳实践：安装与部署

## 安装方法 (Installation Methods)

- **一键安装:** `virtio-win-guest-tools.exe` (含驱动与代理)

Simple, automated installation of all components.

- **安装时加载 (Boot-Time):** 在 Windows 安装阶段加载 `vioscsi` 驱动 (“F6 Driver” 流程)。

Essential for bootable VirtIO storage devices.

## 性能调优 (Performance Tuning)

- **Hyper-V Enlightenments:** 启用 `hv_relaxed`, `hv_spinlocks` 优化时钟与锁机制。

Improves guest OS efficiency under virtualization.

- **Memory:** 配置 CPU Pinning 和 Hugepages 以获得最佳性能。

Reduces latency and increases throughput.

# 未来展望：ARM64 与云原生

## ARM64 Support



- 支持 Windows on ARM (AWS Graviton, Azure Cobalt)。
- Virtio-Win 已增加 ARM64 构建 (Release 0.1.296+)。

## Cloud Native



- OpenShift Virtualization / KubeVirt
- 在容器化基础设施中高效运行 Windows VM 的关键依赖。

# 总结与资源

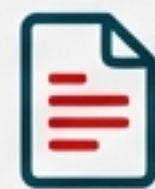
“Virtio-Win 将 Windows 从一个封闭的黑盒转化为 KVM 生态中的高性能一等公民。”



[github.com/virtio-win](https://github.com/virtio-win)



Fedora People  
Repository (Binaries)



Red Hat Customer  
Portal (Docs)