

G A T E D M I G R A T I O N P A T H

Your VMware Exit Plan Assumed a Tool You Didn't Control

The dependency wasn't hidden. The control boundary was.



RACK2CLOUD
Think Like An Architect.
Build Like An Engineer.

Think Like An Architect. Build Like An Engineer.

VMware Exit Plans Map Everything — Except the Tooling

Licensing math

Workload compatibility

Destination platform

Migration schedule

But migration tooling itself is usually treated as a given.

VDDK Didn't Disappear. Access Did.



The dependency remained.

The library still exists. It still does the same job.



The access condition changed.

The public self-service path closed, without notice.

Same tool. Different terms of access.



A Known Dependency Isn't a Controlled Dependency

Your dependency map can be completely accurate — while your organization still has no authority over whether that dependency remains obtainable.

The Control Boundary

YOU CONTROL

Target platform

Migration schedule

Project budget

Destination architecture

Internal staffing

VENDOR MAY CONTROL

Dependency distribution

Access requirements

Library availability

Compatibility path

That Isn't Independence. It's Conditional Permission.

If the tool required to leave the platform depends on the vendor relationship you're trying to eliminate, the exit plan contains a circular dependency.

THE ARCHITECT'S VERDICT

Map who controls your ability to obtain every dependency required to execute the exit.

A tool being documented, tested, and working today does not mean you control its availability tomorrow.

rack2cloud.com | Follow The Architect for more