

Where Each Workload Belongs: Stay, Colocate, or Public Cloud

Getting out of the data center isn't the same as going to the public cloud. Every workload faces three doors — stay, a colocation private cloud, or public cloud. A working guide to placing each one behind the right door.

A PRACTITIONER'S MODEL FOR THE DATA-CENTER EXIT

- Stay & Modernize — your building, your gear, your ops
- Colocate — your private cloud in a rented building (OpenShift, Azure Local)
- Public Cloud — their building, their gear, a shared operating model
- The six signals that decide the door

A data-center exit almost always starts with the wrong sentence: "we're moving to the cloud." That collapses three different decisions into one, and it's how migrations miss their budgets and their reliability targets. "Get out of the owned data center" is a real goal. "Move to the public cloud" is one of several ways to get there — not a synonym for it. The first job of the architect is to keep those two ideas separate, and to make placement a deliberate, per-workload decision across three destinations.

THE DISTINCTION THAT SAVES THE PROGRAM

Colocation is not public cloud. Colo outsources the facility — power, cooling, physical security, carrier-neutral connectivity — while your hardware, and increasingly a private cloud (OpenShift, Azure Local) on it, stay yours. You get the cloud operating model and keep the OS, the data, and the control. Public cloud means the provider's hardware and a shared operating model. Confusing the two turns a facilities decision into a re-architecture bill.

DOOR 1 Stay & Modernize

"Nothing changes but the tech"

Not every workload should leave. Some can't — for data-sovereignty or latency reasons — and some simply shouldn't, because the migration business case is negative. Staying is a legitimate placement, provided you actually modernize in place (virtualize, containerize, refresh) rather than freeze. The one thing "stay" cannot survive is the facility clock: if the owned data center is closing, this door closes with it.

BELONGS HERE

- ◆ Regulated or sovereignty-bound data that can't legally move
- ◆ Ultra-low-latency or mainframe-adjacent systems
- ◆ Workloads where the move's business case is genuinely negative

WATCH OUT FOR

- ✗ "Stay" as an excuse to avoid the decision — the estate never shrinks
- ✗ Staying without modernizing — you keep the debt and the DC cost
- ✗ Choosing "stay" when the facility is actually closing

DOOR 2 Colocate

"A private cloud, in a rented building"

Colocation's core value is outsourcing the data-center facility — backup power, cooling, physical security, and carrier-neutral connectivity — to a specialist who runs it better and cheaper than you can. You keep your hardware; you just stop owning the building full of generators. And modern colo is rarely bare metal: you run a **private cloud** (Red Hat OpenShift, Azure Local) on that hardware, so you also get the cloud operating model — self-service, containers, elasticity within your capacity — while the OS, the data, and the gear stay yours.

BELONGS HERE

- ◆ Wants to offload the facility (power, cooling, security, carriers), not the platform
- ◆ Needs dedicated hardware, appliances, or predictable performance
- ◆ A private-cloud operating model without a public provider

WATCH OUT FOR

- ✗ You still own the hardware lifecycle, capacity, and platform ops
- ✗ Cross-connect and egress costs that erode the savings
- ✗ Treating a stepping-stone as permanent, or vice versa

DOOR 3

Public Cloud

"Everything changes: the operating model"

Public cloud is the biggest change of the three, because it changes how you operate, not just where you run. The provider owns the facility and the hardware; you consume managed services and share the operating model. It rewards workloads that are cloud-ready and earn their keep on elasticity, managed services, and reach. It punishes lift-and-shift: paying cloud prices for data-center architecture is the single most common cost blow-up in a migration.

BELONGS HERE

- Cloud-ready workloads with variable or spiky demand
- New build, and anything you'll genuinely re-platform
- Services that benefit from managed data, AI, or global reach

WATCH OUT FOR

- ✗ Lift-and-shift with no re-platform — cloud cost, DC design
- ✗ Assuming cloud is more reliable by default — it's a different failure model
- ✗ Data gravity and egress; the operating-model change is the hard part

WHAT YOU OWN	STAY	COLOCATE	PUBLIC CLOUD
Building	Yours	Rented	Provider
Hardware	Yours	Yours	Provider
Operations	Yours	Yours	Shared
WHAT CHANGES	NOTHING	THE BUILDING	THE MODEL

The cloud-first trap

A blanket "cloud-first" mandate forces colo-shaped and stay-shaped workloads through a cloud-shaped door — then the program is surprised by the bill and the incident count. **Cloud-first is a preference, not a placement strategy.** The fix is to make each placement a governed decision, based on what the workload actually is.

Choosing the Door

Choosing the door isn't a gut call, and it isn't "cloud by default." Six signals decide where each workload belongs. Weigh them deliberately — the first hard constraint, usually data residency, often closes a door before the rest are even considered.

- 1 Data classification & residency.** Sensitivity and where the data is legally allowed to live. Usually the hardest constraint, so it's decided first — it can close a door before any other factor is weighed.
- 2 Criticality & reliability tier.** How many nines the workload owes. The destination has to be able to deliver that posture — which is where placement hands off to deployment-archetype choice.
- 3 Latency & adjacency.** What has to sit near what. Chatty call chains hate being split across a WAN; adjacency is why dependent apps move as a group, not one at a time.
- 4 Dependencies — the call chain.** The chain moves together, or you split it and pay in latency and new failure modes. Map it before you move it, not after.
- 5 Cloud-readiness.** Can it re-platform, only lift-and-shift, or neither? An honest assessment — not an aspiration — is what separates Door 3 from Door 2.
- 6 TCO of the move.** Does the business case clear, all-in: migration, run cost, egress, dual-running, and the operating-model change? Placement without TCO is a guess.

Governed, not guessed — and a relay, not a rivalry

This isn't a one-time sort. The placement standards run through architecture governance, with conformance measured over time, so the decisions stay honest as the estate changes. It's the difference between designing one building and writing the building code everyone follows.

And which workloads should even go to the cloud — versus colo, versus stay — is an **enterprise-architecture call made before the cloud design starts**. Strong cloud architects and this placement discipline aren't competing; they're a relay.

The Placement Test

Ten checks before you commit a workload to a door. Score one point per honest "yes." Below seven, you don't have a placement strategy; you have a default destination and a hope.

- ☐ Placement is decided **per workload** — not a single default destination for everything.
- ☐ **Data classification and residency are mapped first**, and they constrain the door before anything else is weighed.
- ☐ Every workload has a **reliability tier**, and the chosen door can actually deliver it.
- ☐ **Latency and adjacency are known** — chatty dependencies aren't being split across a WAN by accident.
- ☐ The **full dependency chain is mapped**, and it moves together — or the split is deliberate and costed.
- ☐ **Cloud-readiness is assessed honestly** — re-platform, lift-and-shift, or neither — not assumed.
- ☐ Each move has an **all-in TCO that clears**: migration, run, egress, dual-running, and the operating-model change.
- ☐ **Colo and public cloud are not conflated** — a colo private cloud gives you the operating model and keeps control; a public provider runs the platform.
- ☐ Placement standards are carried through **architecture governance, with conformance**, so decisions stay honest over time.
- ☐ The **facility clock is real and known** — lease and exit timelines drive the sequence, and "stay" is only chosen where it's genuinely an option.

Going Deeper: Frameworks & Companions

The disciplines this model builds on, and two companion notes. All links verified July 2026.

The TOGAF® Standard

The governance and architecture-standards backbone — how placement rules get carried through an ARB and stay conformant.

opengroup.org/togaf

Microsoft — Cloud Adoption Framework

Landing zones and placement guidance for the cloud door — the Door 3 detail this note stays deliberately above.

learn.microsoft.com/azure/cloud-adoption-framework

AWS — The 6 R's of Migration

Rehost, replatform, refactor, and the rest — the migration-strategy vocabulary that lives underneath cloud-readiness.

aws.amazon.com/blogs/enterprise-strategy/6-strategies-for-migrating-applications-to-the-cloud

Uptime Institute

The data-center and colocation reliability lens — tiers, facility resilience, and what "your building vs. rented" really buys.

uptimeinstitute.com

Red Hat OpenShift

An enterprise container / private-cloud platform — one way to run the cloud operating model on your own gear in a colo.

redhat.com/en/technologies/cloud-computing/openshift

Microsoft — Azure Local

Azure's operating model on hardware you own, at your location — the colo private-cloud option in Microsoft's stack.

learn.microsoft.com/azure/azure-local/overview

Companion — You Don't Choose Five Nines

Once you've chosen the door, the reliability tier picks the deployment archetype. Placement and posture, in sequence.

archreviews.com/cloud-playbook/availability-ladder

Companion — Three Views, One Dollar (TBM)

The TCO criterion in depth — how to cost a move honestly, all-in, so the business case isn't a guess.

archreviews.com/field-notes/tbm-cost-transparency

About the Author

Tom Ward is an Enterprise Architect with 20+ years across Fortune 500 financial services, retail, and government — including Wells Fargo and Bank of America — specializing in EA governance, reference architectures, data-center and hybrid-cloud placement, and legacy modernization, with recent hands-on GenAI and mobile delivery work.

More field notes: archreviews.com/field-notes · **Connect:** linkedin.com/in/tomward