CLOUD & INFRASTRUCTURE PLAYBOOK · PART 1

The Five Cloud Deployment Models: Trade-offs and Why Placement Matters More

You're already hybrid multi-cloud. Private, public, hybrid, multi-cloud, community — their real trade-offs, and why the models are just vocabulary while placement is the work.

DEPLOYMENT MODELS — A PRACTITIONER'S PRIMER

- **Private — exclusive to you, owned DC or colo**
- **Public — provider-run and open to all**
- **Hybrid — private + public, bound together**
- **Multi-cloud — two or more public providers**
- **Community — shared by orgs with common compliance**

Tom Ward
Enterprise Architect · Cloud & Infrastructure

linkedin.com/in/tomward
July 2026

Every cloud strategy conversation starts with the same argument: public, private, hybrid, or multi-cloud? It's the wrong argument. If you run a large enterprise, you're already all of them — regulated workloads on private infrastructure, elastic ones on public, and almost certainly more than one public provider, arrived at by acquisition or by a team picking the best tool for a job. That isn't a model you chose. It's a condition you're in. The models are just vocabulary for where things already run; the work is deciding what should run where. First, the vocabulary — honestly.

THE TAXONOMY, STRAIGHT

Four of these come from NIST's definition of cloud (private, community, public, hybrid); multi-cloud is the industry's fifth. The common confusion worth clearing: hybrid mixes private and public; multi-cloud means two or more public providers. The real end-state for most large banks is both at once — hybrid multi-cloud.

Private

exclusive to one org — owned DC or colo

Buys: control, isolation, data residency, predictable performance — and with OpenShift or Azure Local, the cloud operating model on your own gear. **Costs:** you own the capex, capacity, refresh, and platform operations, and your elasticity is bounded by what you built.

Public

provider-run, open to all — AWS, Azure, GCP

Buys: elasticity, managed services, global reach, and no capital outlay — the fastest path to new capability. **Costs:** less control, shared responsibility, data-residency and egress constraints, lock-in, and cost that surprises at scale if left ungoverned.

Hybrid

private + public, bound together

Buys: the right workload in the right place — regulated on private, bursty on public — plus a migration bridge and cross-environment resilience. **Costs:** complexity at the seam: networking, identity, data gravity, and keeping governance and security consistent across both sides.

Multi-cloud

two or more public providers at once

Buys: the ability to survive one provider's failure, freedom from lock-in, best-of-breed services per cloud, and negotiating leverage. **Costs:** the highest operational complexity of any model — portable architecture, duplicated skills and tooling, cross-cloud identity and networking, inter-cloud egress, and the "lowest-common-denominator" trap of only using features common to all.

Community

shared by orgs with common needs

Buys: shared cost and a shared compliance baseline for a sector or consortium. **Costs:** governance by committee and limited flexibility — real, but niche in banking, where most institutions have the scale to run their own private estate.

The honest note on multi-cloud

Multi-cloud is the most oversold of the five. **Most "multi-cloud" isn't resilience multi-cloud.** True active-active across providers needs a portability that most applications simply don't have — the research literature still calls it "in its infancy."

What enterprises actually have is **best-of-breed** multi-cloud (this workload runs better over there) or **by-acquisition** multi-cloud (we inherited a company that ran on the other one). Naming that difference honestly saves a great deal of wasted architecture — and a great deal of money spent chasing a resilience story the design can't deliver.

None of this is a menu you order from once. A large enterprise runs several of these at the same time, on purpose in some places and by accident in others. Which is exactly why the model is the wrong thing to argue about — and the right thing to argue about is one layer down.

The Model Is a Label. Placement Is the Work.

A deployment model is just a name for where a workload already runs. It tells you almost nothing about whether it's in the right place. The decisions that actually determine cost, risk, and reliability sit one layer below the model — and there are three of them.

1. **Where each workload should go** — stay in place, move to a colocation private cloud, or go to public cloud. This is the *environment* decision, made per application by data classification, criticality, latency, dependencies, cloud-readiness, and TCO. Get it wrong and you pay cloud prices for data-center architecture, or you strand a workload that should have moved.
2. **Which zone it lands in** — the *neighborhood*, defined by risk, security, data privacy, and criticality. The same map applies in every environment, so this is what lets a migration keep its security posture instead of quietly weakening it.
3. **How reliable it must be** — the *deployment archetype*, from zonal to global, that delivers the workload's target availability. The nines you owe decide the posture you buy.

Get those three right and the model takes care of itself

Argue about the model — "should we be multi-cloud?" — and every hard decision is still ahead of you. Decide placement, zone, and reliability per workload, and the "model" is simply the sum of where everything landed. **The label describes the outcome; it was never the decision.**

Stop debating the model. Start placing the workload.

The Model Reality Test

Ten checks on whether your organization is running its deployment models on purpose. Score one point per honest "yes." Below seven, your "cloud strategy" is a collection of accidents with a label on top.

- ☐ You can name **which models you actually run today** — and which arrived by acquisition or by a team's local choice.
- ☐ You treat **hybrid multi-cloud as the state you're in**, not a future decision to be made.
- ☐ You know the difference between **hybrid** (private + public) and **multi-cloud** (two or more public) — and use the terms precisely.
- ☐ Your **"private" includes colocation private cloud** (OpenShift, Azure Local), not just an owned data center.
- ☐ Your multi-cloud is honestly labeled — **best-of-breed, by-acquisition, or genuine resilience** — and priced accordingly.
- ☐ You are not paying for a **resilience story your architecture can't deliver** (active-active across clouds the apps can't support).
- ☐ Placement is decided **per workload** — environment, neighborhood, reliability — not by a blanket model mandate.
- ☐ Governance and security are **consistent across the seams** between private and public, and between clouds.
- ☐ **Cost is visible across every environment**, so the "which model" question can be answered with numbers.
- ☐ New workloads get a **deliberate placement** at design time — so the estate grows on purpose, not by default.

Going Deeper: Reference & the Placement Arc

The definition behind the vocabulary, and the three companion notes where the real work happens. All links verified July 2026.

NIST SP 800-145 — The NIST Definition of Cloud Computing
The canonical source for the deployment models (private, community, public, hybrid) and the service models. Start here.
csrc.nist.gov/pubs/sp/800/145/final

The TOGAF® Standard
The governance backbone that turns a scatter of models into a deliberate, per-workload placement discipline.
opengroup.org/togaf

Companion — You Don't Choose Five Nines
The reliability layer: how a deployment archetype delivers the nines a workload owes.
archreviews.com/cloud-playbook/availability-ladder

Companion — Three Views, One Dollar (TBM)
The cost layer: how to price a placement honestly, all-in, so "which model" can be answered with numbers.
archreviews.com/field-notes/tbm-cost-transparency

The placement arc — four notes, one decision stack

Models (this note) → **Environment** (stay / colo / public — the "Three Doors") → **Neighborhood** (the security zone within an environment) → **Reliability archetype** (the nines). The model is where you end up; these three are how you get there deliberately.

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