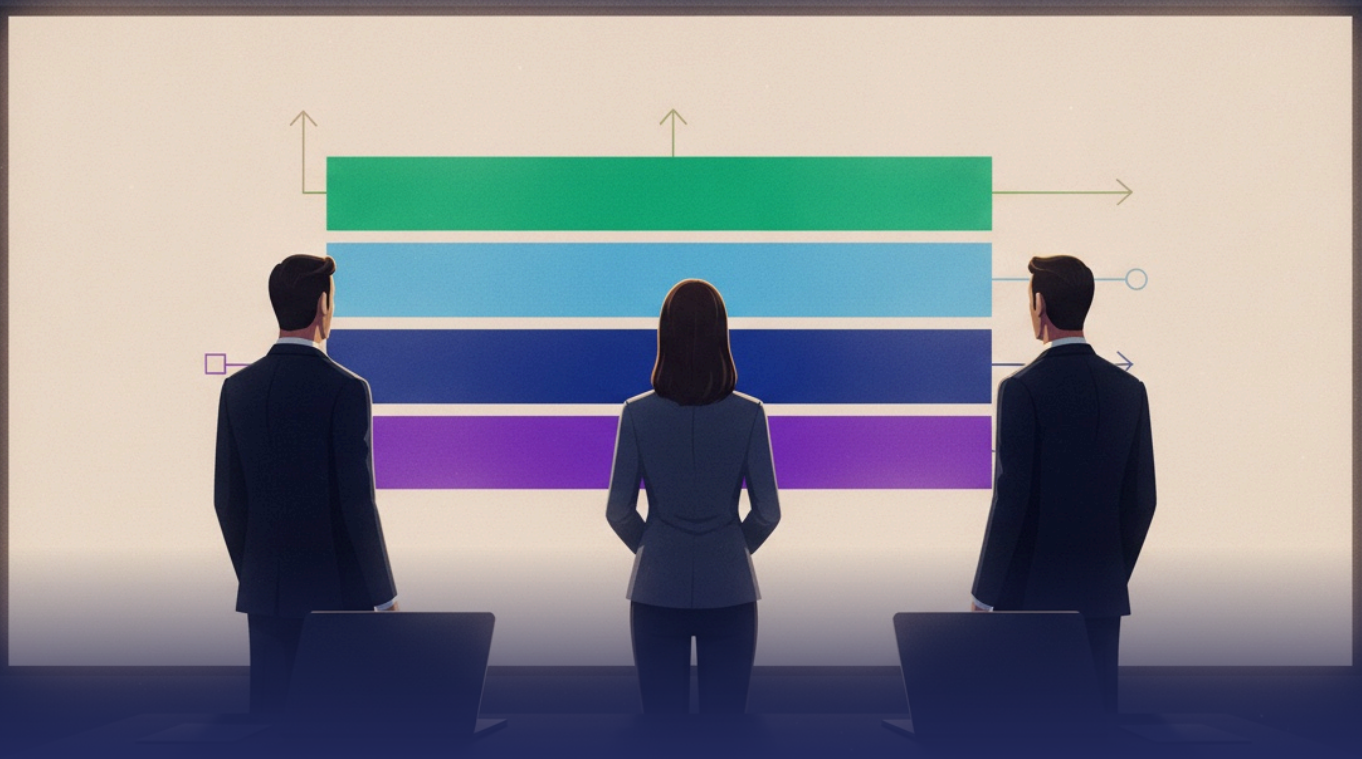

FIELD NOTE · IT ECONOMICS · 01

Three Views. One Dollar.

The four-layer cost taxonomy that gets Finance, IT, and the business to stop arguing about the math — and the allocation ladder that decides whether anyone believes the answer.

A WORKING GUIDE TO THE TBM UNIFIED MODEL

- **Business Units & Capabilities — who consumes it?**
- **Applications & Services — what do we deliver?**
- **IT Towers & Sub-Towers — what technology functions?**
- **Cost Pools & Sub-Pools — what did we actually buy?**
- **The Allocation Ladder — assumption → attribute → consumption**

Tom Ward
Enterprise Architect · Cloud & Infrastructure

linkedin.com/in/tomward
July 2026

Finance has a number, IT has a different number, and the business has neither. All three are looking at the same dollar in three different languages. The TBM Taxonomy resolves this the way GAAP resolved it for accounting — with a shared structure nobody relitigates each quarter. Four layers; every dollar enters at the general ledger and routes upward through all of them.

LAYER 1 Cost Pools & Sub-Pools

"What did we actually buy?"

The bottom layer, closest to the ledger you already have. Costs land in a small set of pools — internal and external labor, outside services, hardware, software, facilities & power, telecom, internal services — each split into expense and capital sub-pools. This is the **Finance view**: terms a controller recognizes without translation.

The value is in how specific the definitions are. The Hardware Expense sub-pool covers non-capitalized purchases — spare parts, consumables, equipment below the capitalization threshold — and nothing else. That precision is the point: it stops two teams from booking the same cost in two places and calling both correct.

KEY DECISIONS

- Does your chart of accounts map cleanly to the standard pools, or does it need a translation table?
- Where do the expense/capital boundaries sit, and are they consistent across regions?
- Who arbitrates an ambiguous mapping — and is that decision written down?

WATCH OUT FOR

- × Mapping rules that live in one analyst's head and leave when they do.
- × "Other" quietly becoming the dumping ground for anything contested.
- × Capitalization thresholds that differ by entity, silently skewing comparisons.

LAYER 2 IT Towers & Sub-Towers

"What technology functions?"

Where Finance's language becomes IT's. The standard towers — data center, compute, storage, network, platform, end user, application, delivery, security & compliance, IT management — describe technology *functions*, each breaking into concrete sub-towers.

The Windows Compute sub-tower, for example, is everything associated with servers running Windows Server: hardware, software, labor, and support. That all-in framing is deliberate — a unit cost is comparable to a peer's only if both sides agree what is inside the unit. It is also why this layer is where benchmark data attaches.

KEY DECISIONS

- Which towers do you actually operate, and which sub-towers carry material spend?
- What is your unit of measure per tower, and can you count it reliably today?
- Do you intend to benchmark? If so, resist customizing this layer.

WATCH OUT FOR

- × Inventing custom towers before exhausting the standard ones.
- × Towers that mirror the org chart — the next reorg breaks the model.
- × A sub-tower defined as "hardware only" while the benchmark assumes all-in.

LAYER 3 Applications & Services

"What do we deliver?"

The first layer a business partner recognizes without a translator. It separates **business services** — product management, sales & marketing, manufacturing & delivery, customer service — from **shared services** like finance, workforce, procurement, risk & compliance, legal, and corporate communications. Alongside them sit the IT-facing service groups: end user, delivery, platform, infrastructure, and emerging.

That last group is worth noticing. "Emerging" is where the taxonomy parks IoT, virtual reality, and blockchain — and where anyone building this model today will find themselves filing GPU capacity, model inference, and token spend. It is the clearest sign that a taxonomy is a living artifact, not a monument.

KEY DECISIONS

- Is there one authoritative application inventory, and does it agree with the CMDB?
- Do you cost an application all-in — buy, build, and run together?
- Where does AI and cloud-native spend land before "emerging" stops being honest?

WATCH OUT FOR

- × An app inventory that disagrees with the CMDB, so both get distrusted.
- × Services defined by who owns them rather than what is delivered.
- × Shadow applications absent from the model but present in the invoice.

LAYER 4 Business Units & Capabilities

"Who consumes it?"

The top layer, where the CFO's dollar finally has a name: originations, claims, fulfillment, a channel, a product line. The taxonomy allows two shapes — consuming **business units** and enabling **business capabilities** — and most organizations need both, because "who paid for it" and "what it enabled" are different questions.

This is the layer where TBM stops being a costing exercise and starts being a conversation about value. It is what makes the questions leaders actually ask answerable: IT cost per employee by business unit, total cost to buy-build-run a given application, the run-the-business versus innovation ratio, and the fixed-versus-variable split that determines how fast you could cut if you had to.

KEY DECISIONS

- Business units, capabilities, or both — and who owns each definition?
- Do you *show back* cost or *charge back* for it? They demand different rigor.
- Which three questions must this layer answer in its first quarter?

WATCH OUT FOR

- × Publishing chargeback before the allocations are defensible. Credibility is spent once.
- × A capability model invented for TBM that no one else in the enterprise uses.
- × Reporting so granular that no one can act on any single line of it.

The Allocation Ladder

The taxonomy tells you which buckets exist. It does not tell you how a dollar moves between them — and that is where cost models lose their credibility. Three routing strategies are available at every hop. They are presented as options; they behave like a maturity ladder, because each rung is harder to argue with than the one below it.

Rung 1 Assumption-Based

FASTEST · WEAKEST

Cost is routed on a stated assumption. It requires no new data, which is exactly why nearly every model starts here — and why so many never leave.

In practice: labor split 25/75 across Wintel and Unix compute · data center cost estimated from a rate card at $50/kW-month · application support labor spread evenly across applications · business application cost allocated across lines of business by percentage of revenue.

Rung 2 Attribute-Based

WEIGHTED BY SOMETHING REAL

Cost is weighted by an attribute of the thing itself. You are no longer asserting a split — you are deriving it from a property someone can independently verify.

In practice: data center cost weighted by CPU count or kW power rating · desktop cost weighted by make and model · application support labor weighted by application size or complexity · business application cost weighted across lines of business by number of assigned login accounts.

Rung 3 Consumption-Based

METERED · HARDEST TO DISPUTE

Cost follows measured consumption. This is the rung where a business unit leader can change their bill by changing their behavior — which is the entire point of cost transparency.

In practice: data center cost allocated on power actually consumed during the month · server cost allocated to applications by total compute hours · application support labor allocated by support tickets · business application cost allocated by number of business transactions.

The honest approach is a **mix**, chosen per allocation and matched to the data you have. Rung 3 everywhere is a fantasy; rung 1 everywhere is why nobody trusts the chargeback. What matters is knowing which rung each allocation stands on — and saying so before someone else asks.

The configure-versus-customize trap

Configuration — adding an industry-specific tower, say — keeps compliance with the standard, keeps upgrades straightforward, and keeps your numbers comparable to benchmark data. **Customization** reshapes the model to match how your organization already thinks, and quietly forfeits all three. Both are permitted; only one is cheap later.

Apptio reports over half its customers run the standard model — including many who built a custom one first and worked their way back. The oldest lesson in enterprise architecture, in a finance hat: **the value of a standard is not that it is optimal, it is that it is shared.**

The Cost Transparency Baseline

Ten checks that separate a cost model people act on from a spreadsheet people dispute. Score one point per true statement. Below seven, fix the gaps before you publish a number to a business unit leader — you get one chance at that credibility.

- ☐ **Every IT dollar has one path** from the general ledger to a business unit or capability, and you can trace an individual cost end to end.
- ☐ **The taxonomy is the standard one** — you have configured within it rather than customized around it, and you can still compare to benchmark data.
- ☐ **Each allocation's rung is known and documented** — assumption, attribute, or consumption — and published alongside the number, not hidden behind it.
- ☐ **The data sources are named and owned** — GL, fixed assets, CMDB, PPM, HR, service desk — with a human accountable for each feed's accuracy.
- ☐ **Cost pool mapping rules are written down**, reviewed, and survive the departure of the analyst who wrote them.
- ☐ **Application inventory and CMDB agree**, or the discrepancy is a known, tracked, shrinking number.
- ☐ **Unit costs are defined all-in** — hardware, software, labor, and support — so a peer comparison compares the same thing.
- ☐ **You can answer the four executive questions**: cost per employee by business unit, total cost to buy-build-run an application, run versus innovation split, and forecast versus plan by cost pool.
- ☐ **At least one allocation has climbed a rung** in the last year, and the improvement was visible to the business unit it affected.
- ☐ **Someone outside IT has challenged the model** and the answer held up — a cost model that has never been contested has never been tested.

Going Deeper: References

Where to go next, roughly in the order a team standing up cost transparency should read them. All links verified July 2026.

The TBM Council
The nonprofit standards body behind the taxonomy. Start here — it is the vendor-neutral home of the model.
tbmcouncil.org

The TBM Taxonomy (current release)
The four-layer model itself, maintained past the 2020 white paper this note draws on.
tbmcouncil.org/learn-tbm/tbm-taxonomy

Learn TBM — practitioner resources
The Council's resource center: adoption guidance, case material, and vertical workgroup output.
tbmcouncil.org/learn-tbm

IBM — What is Technology Business Management
A plain-language primer, useful for briefing an executive who has not met the discipline before.
ibm.com/think/topics/technology-business-management

Apptio (now part of IBM)
The vendor whose unified model this note analyzes — worth knowing the tooling even if you never buy it.
ibm.com/products/apptio

Apptio resource library
Where the ATUM white paper and its companion material live.
apptio.com/resources

FinOps Foundation — the Framework
TBM's cloud-era sibling. Same instinct, applied to spend that changes hourly rather than annually.
finops.org/framework

What is FinOps
The shortest path to understanding how cloud cost management relates to, and differs from, TBM.
finops.org/introduction/what-is-finops

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, hybrid cloud, and legacy modernization, with recent hands-on GenAI and mobile delivery work.

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