CHAPTER 1 / 4 MINUTE BRIEF

What Computing Costs

Read the full chapter → / Read this brief online →

The main idea

Go back to the case from the introduction: a demonstration has turned into a launch, the service can absorb the new work only while nothing fails, and the additional capacity won't arrive in time. One of Finance's first questions is why the resources we already pay for can't do more of the work. It's a fair question. Before we can answer it, or ask anyone to approve a purchase, we need to say what the organization pays for today, what it has already promised to keep paying, and what would actually change if we did something different.

Explore this section in the full chapter →

Knowing what you consume before you change it

Sometimes an application will stop using resources the organization has already committed to paying for. Releasing them may let another team use them or avoid a purchase later, but it does not cancel the payment. Every proposed change has to be followed through to the obligation before anyone calls it a reduction.

Explore this section in the full chapter →

Several dimensions of the same dollar

Fixed, variable, committed, metered: these are useful words, but trouble starts when we treat them as four separate boxes and assume that placing an expense in one tells us how to manage it. A single cloud product might charge one component as fixed overhead and another by usage. Consumption can be metered whether it is covered by a commitment or billed at an on-demand rate. A service whose usage varies can still carry a minimum payment. The label alone does not tell us how the charge will respond to a change in use.

Explore this section in the full chapter →

Then there is **attribution**, which connects an expense to the workload or decision that caused it. The easy version is to tag resources to the teams that use them and charge those teams, but many costs are shared. Some organizations spread them evenly across the company and call it a *peanut butter spread*; management accountants know it as peanut-butter costing and warn that it overcharges light users and undercharges heavy ones.[1] Other shared costs go to the groups that use a particular service, and some are simply overhead. Attribution also gets political, because who pays is a question about who is answerable. It tells us who is consuming resources and where to start a conversation, not what the enterprise would save if that consumer went away, and shifting a cost between teams is surprisingly easy to mistake for reducing it.

Explore this section in the full chapter →

[1] Charles T. Horngren, Srikant M. Datar, and Madhav V. Rajan, *Cost Accounting: A Managerial Emphasis*, 15th ed. (Pearson, 2015), chapter 5, "Activity-Based Costing and Activity-Based Management": broad averaging, or peanut-butter costing, spreads resource costs uniformly across products or services that use those resources non-uniformly, overcosting some and undercosting others. The application to shared cloud costs is the author's.

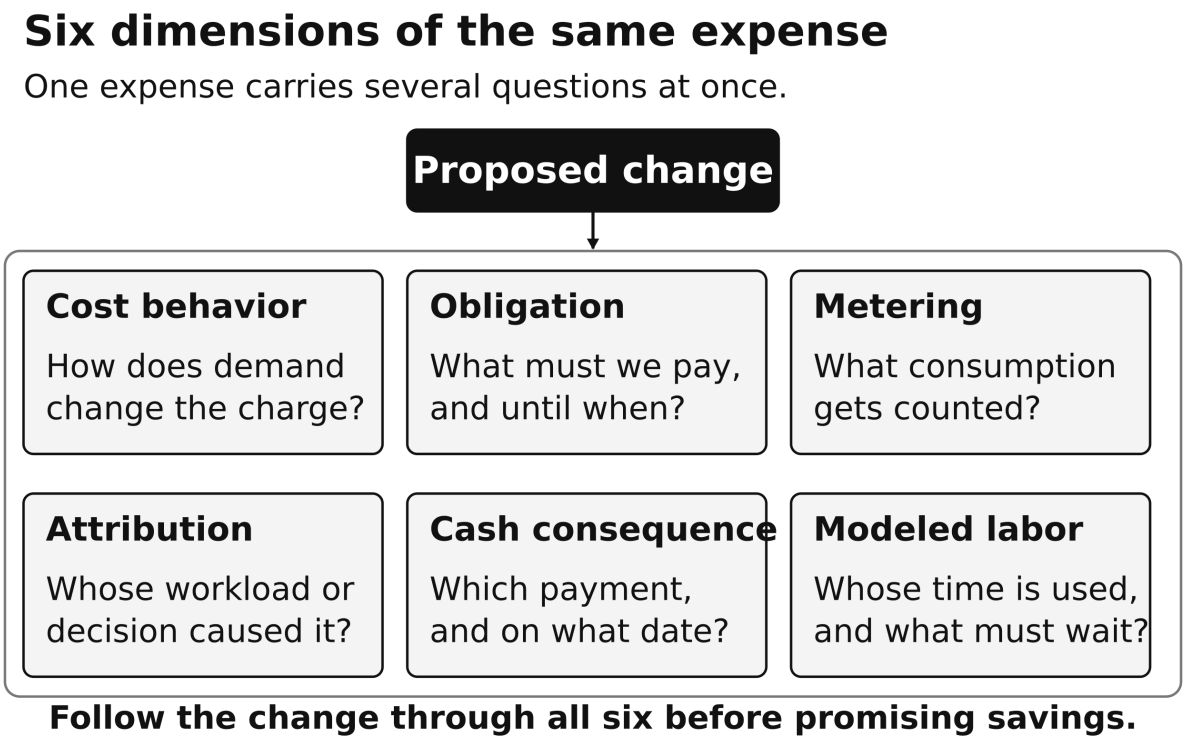


Diagram 1.1. Every expense carries overlapping dimensions: how its charge behaves, what is owed, what is counted, who caused it, which payment changes, and whose time is consumed. These questions must be answered together.

Explore this section in the full chapter →

A small number of teams dominate cost

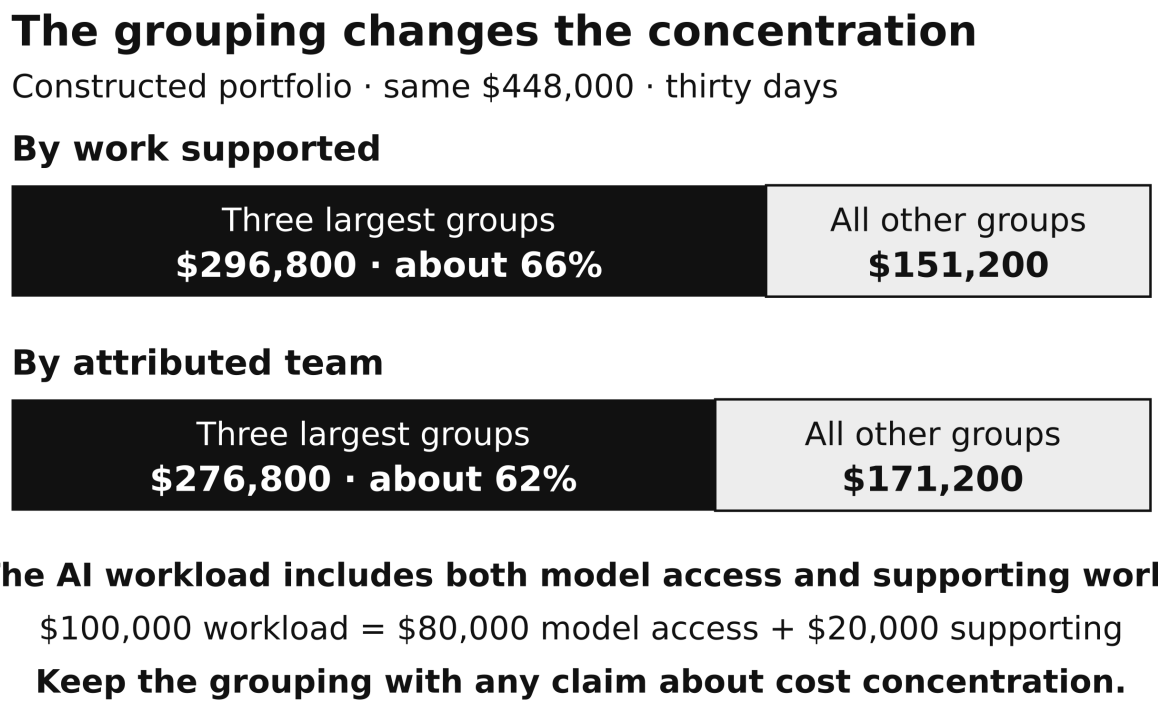


Diagram 1.2. The three largest workloads account for $296,800 of the $448,000 bill, about 66 percent; the three largest attributed teams account for $276,800, about 62 percent. The difference is the AI workload's supporting work, which belongs to other teams. Keep the grouping with the concentration percentage.

Explore this section in the full chapter →

How the charges behave

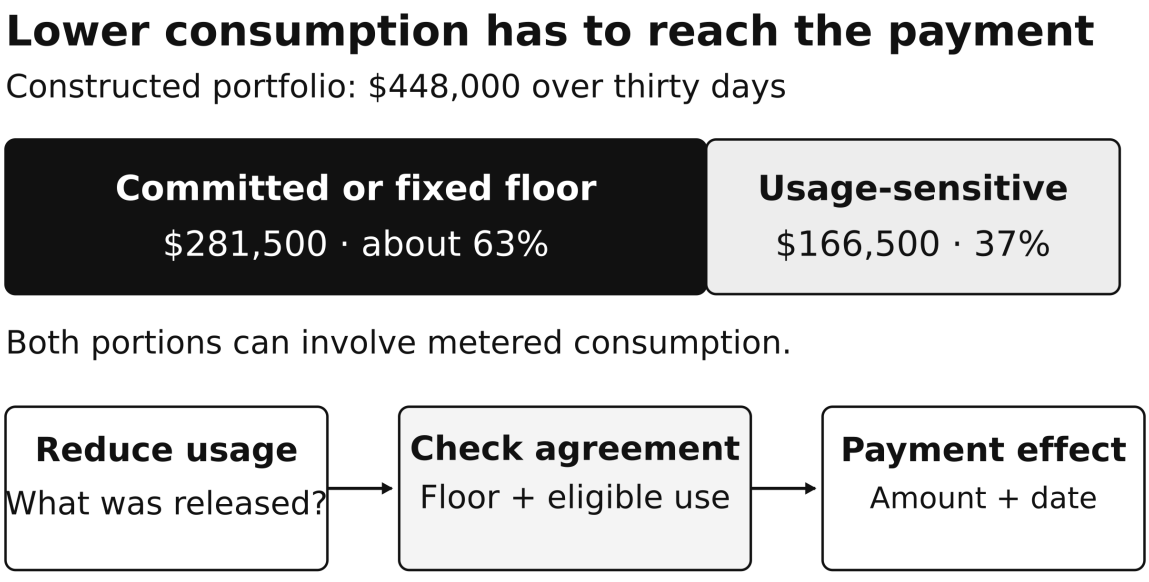


Diagram 1.3. In the constructed thirty-day portfolio, $281,500 is a committed or fixed floor and $166,500 is usage-sensitive. Reducing consumption must be followed through the agreement before it becomes a payment reduction; seasonality is a separate property of demand.

Explore this section in the full chapter →

Follow a change across the boundary

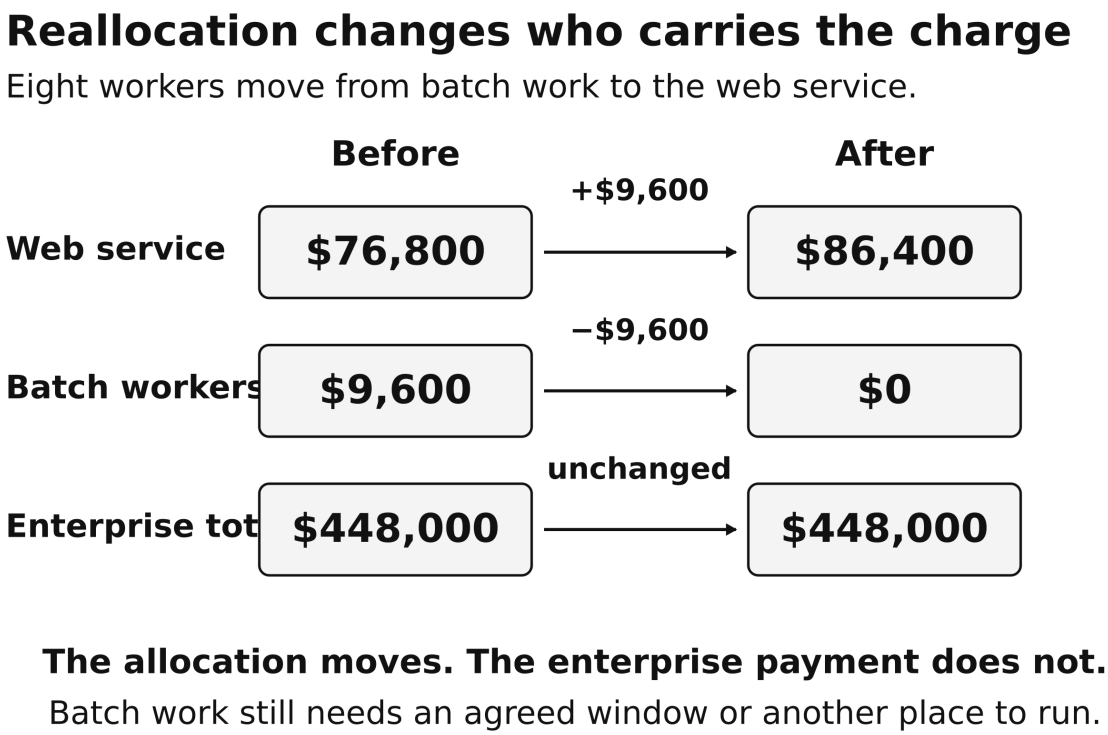


Diagram 1.4. Moving eight batch workers changes attribution: the web service rises by $9,600, the batch allocation falls by $9,600, and the enterprise total stays $448,000. The batch work and any funding transfer still need separate agreement.

Explore this section in the full chapter →

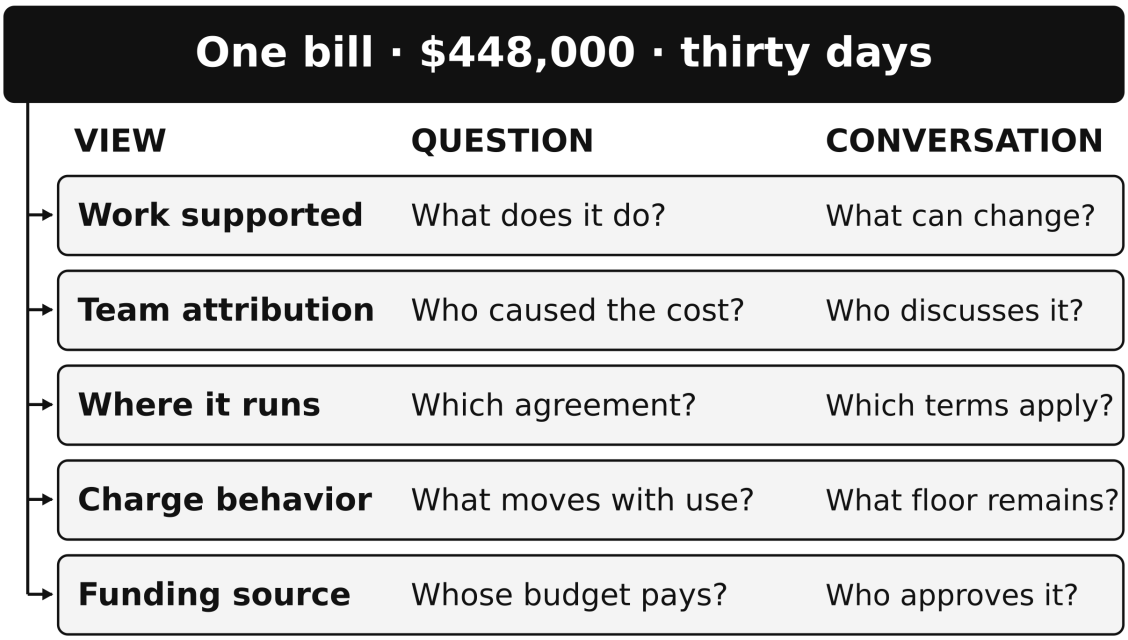
The same bill, five ways

The same $448,000 has now appeared as five tables, each because a different conversation needs it. The work view says what could change and what the change would disturb; the attribution view, whom to talk to; the location view, which agreement governs the change; the charge view, how much of the bill is floor; and the funding view, whose budget moves and whether its owner has agreed.

Explore this section in the full chapter →

One bill, five conversations

Each view groups the same money; never add these totals.



A proposal needs all five conversations about the same expense.

Diagram 1.5. The same $448,000 over thirty days grouped five ways: by work supported, team attribution, where it runs, charge type, and funding source. Each view answers a different question about the same money.

Explore this section in the full chapter →

Enough information to have the next conversation

I would begin the cost record with the work being supported, its owner, the period, the amount, the decision under consideration, and whether each charge is direct, shared, or still unresolved, then follow the change through every dimension. For this chapter's two changes, it would start like this:

Dimension	Eight borrowed batch workers	Sixteen new workers
Amount	$9,600 per thirty days moves to the web service; total unchanged	$19,200 per thirty days added, before any commercial offset
Cost behavior	Usage-sensitive; the charge continues because the workers stay in use	Usage-sensitive unless bought under a commitment
Obligation	None new; the agreement with the batch owner sets the return date	None until bought; a one- or three-year commitment would outlast the launch
Metering	Worker-days, unchanged	Worker-days from provisioning, which precedes qualified service
Attribution and funding	Web service; funding moves only if both budget owners agree	Web service; website product line, if it agrees
Cash	No additional payment	Set by the agreement, effective date, and invoice terms
Modeled labor	Qualifying the moved workers; the batch catch-up	Qualifying the new supply
Still open	The cost of the batch delay: unpriced, not zero	Lead time to qualified service

Keep the underlying records available, because the first version will contain questions that somebody else can answer better than we can.

Explore this section in the full chapter →