European Journal of Computer Science and Information Technology, 14(4),70-91, 2026
Print ISSN: 2054-0957 (Print)
Online ISSN: 2054-0965 (Online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development -UK

Where the Real Money Hides: A Multi-Layer Cost Anatomy of Enterprise Database Workloads in the Cloud

Devendra Rajput
Technical Architect Manager, Accenture, Richmond, Virginia, USA

doi: https://doi.org/10.37745/ejcsit.2013/vol14n47091 Published June 28, 2026

Citation: Rajput D. (2026) Where the Real Money Hides: A Multi-Layer Cost Anatomy of Enterprise Database Workloads in the Cloud, *European Journal of Computer Science and Information Technology*, 14(4),70-91

***Abstract*—*Enterprise cloud cost optimization research has matured around stateless, ephemeral compute workloads, leaving a structural blind spot in the literature: the cost behavior of stateful, license-bound enterprise database workloads. Industry data indicates that 21–40 percent of enterprise cloud spend is wasted, yet existing FinOps frameworks and academic surveys treat compute as the primary cost vector and abstract away licensing, replication, audit, observability, and architectural-debt costs that often dominate the total bill for the database tier. This paper introduces a multi-layer cost anatomy of enterprise database workloads in the cloud, comprising ten interacting cost layers that collectively account for the majority of unrecognized spend. We formalize a License–Performance–Cost trilemma showing why local optimization on any single axis frequently produces global cost increases. Drawing on practitioner experience from large-scale Oracle Exadata, Real Application Clusters, Oracle Cloud Infrastructure, and Exadata Cloud@Customer deployments, we present an empirical decomposition methodology, a reference instrumentation architecture spanning Fleet Patching and Provisioning (rhpctl), the database administration CLI (dbaascli), Enterprise Manager, and native cloud telemetry, and an illustrative case study decomposing the cost of a representative migration. We further discuss the implications of recent regulatory developments including the European Union Data Act cloud switching rules and the Digital Operational Resilience Act for cost governance, and the under-explored carbon–cost–license interaction. The paper closes with a research agenda highlighting eight open problems in which database-aware FinOps differs structurally from generic cloud cost optimization.*

Keywords: Cloud cost optimization, FinOps, Oracle Database, Exadata Cloud@Customer, Bring Your Own License, database licensing economics, cost anomaly detection, cloud sustainability, regulatory compliance, enterprise architecture.

INTRODUCTION

Cloud computing was sold to the enterprise on a deceptively simple proposition: convert capital expenditure into elastic operating expenditure aligned with workload demand. A decade and a half into

widespread enterprise adoption, the dominant question has shifted from whether to migrate to how to control the resulting spend. Industry surveys consistently report that managing cloud cost is the single largest reported challenge in enterprise IT, with approximately one-fifth to two-fifths of all cloud expenditure classified as waste depending on the methodology applied [1], [2], [3].

A substantial body of research and a vibrant practitioner community have emerged in response. The FinOps Foundation now frames cloud financial management as a recognized discipline with maturity models, capability domains, and a vendor ecosystem of dedicated tools [4]. Academic surveys have catalogued machine-learning-based resource allocation [5], [6], heuristic and meta-heuristic placement algorithms, graph-based cost modeling [7], and integrated frameworks combining predictive scaling with reactive heuristics [8]. Industry venture capital has documented at scale how cloud margin pressure can erode public-company market capitalization, popularizing the contested concept of cloud repatriation [9].

Despite this maturity, the existing literature exhibits a persistent and consequential blind spot. The dominant unit of analysis is the stateless, ephemeral, horizontally-scalable workload the web tier, the API service, the batch job, the Kubernetes pod. This bias reflects both the publish-friendly availability of public benchmark traces and the reality that such workloads admit clean optimization formulations. Yet for a large class of enterprise environments particularly those running mission-critical online transaction processing, data warehousing, enterprise resource planning, and core banking systems the largest single cost center is not stateless compute but the database tier, which is structurally different in every dimension that matters for cost optimization.

Database workloads, especially those built on engineered systems such as Oracle Exadata and on managed services such as Exadata Cloud Service and Exadata Cloud@Customer, exhibit at least six properties that fundamentally invalidate the assumptions of generic cloud cost optimization:

- They cannot generally run on spot or preemptible instances, removing one of the most-cited cloud cost levers in the literature.
- They carry license costs that often exceed infrastructure costs by a factor of two or more, making the dominant FinOps metric cost per OCPU-hour or vCPU-hour a misleading proxy for total cost.
- Their license metrics are tied directly to physical or virtual core counts. Performance tuning decisions therefore modulate license consumption, creating a feedback loop that is absent from stateless-compute models.
- They maintain redundancy through Real Application Clusters, Data Guard standbys, GoldenGate replicas, and multi-region copies that are non-elastic and incur duplicate cost across all of the above dimensions.
- They generate substantial auxiliary cost in backup, audit, observability, and replication that is rarely present in published academic models and rarely surfaced as a first-class category in FinOps tooling.

- They embed architectural decisions taken years before the cloud migration that may now be financially irreversible without major refactoring, creating a category of locked-in waste we term architectural debt.

The consequence is that a large share of enterprise cloud waste resides in the database tier, hidden from the dashboards FinOps teams are taught to read and absent from the optimization formulations academic researchers are taught to write. The present paper addresses this gap directly.

Research Questions

This paper is organized around three research questions:

- What is the appropriate unit of analysis for cost optimization of enterprise database workloads in the cloud, and how does it differ structurally from the stateless-compute formulations dominant in literature?
- Can the hidden cost layers in enterprise database cloud deployments be decomposed into a stable, generalized taxonomy that supports both academic study and practitioner instrumentation?
- What are the interactions among license metrics, performance tuning, and cloud sizing decisions, and under what conditions local optimization on one axis produce a global cost increase?

Contributions

The paper makes four contributions: a multi-layer cost anatomy for enterprise database cloud workloads comprising ten formally defined cost layers (Section 2); a formal License–Performance–Cost trilemma model demonstrating the conditions under which conventional rightsizing increases total cost in license-bound workloads (Section 2); an empirical decomposition methodology and a reference instrumentation architecture spanning Oracle Fleet Patching and Provisioning, the database administration CLI, Enterprise Manager, and native cloud cost telemetry, mapped onto each cost layer (Section 3); and a practitioner-grounded case study decomposing the cost of a representative migration of an Oracle Real Application Clusters estate to a hybrid Exadata Cloud@Customer and Oracle Cloud Infrastructure topology (Section 4).

Paper Organization

The remainder of the paper is organized as follows. Section 2 reviews the relevant literature and presents the theoretical underpinning of the study the ten-layer cost taxonomy and the License–Performance–Cost trilemma. Section 3 describes empirical methodology and the reference instrumentation architecture. Section 4 reports the results, including the inverted ranking of cost layers and the migration case study. Section 5 discusses regulatory, sustainability, and repatriation implications. Section 6 develops the implications for research and practice. Section 7 concludes, and Section 8 sets out a future-research agenda of eight open problems.

Literature and Theoretical Underpinning

This section first reviews the relevant literature across five threads and positions the present work within it, then develops the two theoretical constructs on which the empirical study rests: a ten-layer cost anatomy and a License–Performance–Cost trilemma.

Cloud Cost Optimization Surveys

Comprehensive reviews of cloud cost optimization have appeared periodically. Deochake [10] surveys cost optimization strategies including reserved capacity purchasing, spot instance utilization, autoscaling, and architectural choices, with case studies from large public-cloud-native organizations such as Pinterest and Netflix. Mansouri et al. [11] focus on cost optimization for cloud storage from the user perspective, providing one of the few systematic taxonomies of storage-tier cost. Recent reviews extend coverage to artificial intelligence infrastructure, graphics processing unit economics, and Kubernetes-native cost tooling [10]. These surveys are valuable but adopt the stateless-compute frame; the database tier is treated either as out of scope or as a generic compute workload, and the licensing dimension is largely absent from their formulations.

Resource Allocation and Scheduling

A large body of work optimizes virtual machine placement, container scheduling, and workload allocation under cost objectives. Boghani et al. [7] propose a graph-based convex optimization framework addressing limitations of the Kubernetes Cluster Autoscaler. Wang and Yang [6] combine long short-term memory demand prediction with deep reinforcement learning for dynamic scheduling. Hybrid frameworks [8] integrate predictive and heuristic methods with reported cost reductions of 30 percent or more on synthetic workloads. None of these approaches admit license metrics into the formulation, which makes them inapplicable to license-bound workloads where the dominant cost gradient is not in the variable they optimize.

FinOps Practice and Cost Allocation

The FinOps Foundation has formalized maturity models, capability frameworks, and a regular reporting cadence on industry practice [4]. Practitioner work has examined chargeback versus showback economics, real-time cost visibility, tagging practice, and the persistent disconnect between FinOps teams and engineering [3], [12]. This thread is rich on organizational and process aspects but generally model-free, and it largely treats license costs as exogenous parameters rather than decision variables.

Cloud Repatriation and Architectural Debt

Wang and Casado [9] argued that for software companies operating at scale, public-cloud margin compression destroys substantial market capitalization, with reported repatriation savings ranging from one-third to one-half of equivalent cloud spend. Subsequent industry analysis has tempered this conclusion with a more nuanced hybrid view, acknowledging that lift-and-shift workloads repatriate cleanly but cloud-native architectures incur significant refactoring cost on the way back. The thread is largely

qualitative and does not formalize the cost components that drive the repatriation calculus, leaving practitioners with a binary framing of a question that is fundamentally a portfolio problem.

Cloud Sustainability

A growing body of work integrates carbon emissions into cloud cost models. Farrahi Moghaddam and Cheriet [13] propose a virtual carbon tax to internalize environmental cost into cloud profit optimization. Recent work integrates carbon-aware scheduling into artificial intelligence inference and training workloads. To our knowledge, the joint interaction of sustainability constraints, regulatory data residency constraints, and license terms the three exogenous constraints that practically dominate enterprise database region selection has not been formalized in the cost optimization literature.

Positioning

We extend the literature in four directions: first, we shift the unit of analysis from stateless compute to license-bound stateful workloads; second, we decompose the cost vector into ten layers rather than a single compute-dominant scalar; third, we formalize the licensing axis as a first-class constraint rather than an exogenous parameter; and fourth, we integrate regulatory and sustainability considerations into the cost model in a way that preserves their structural differences from purely commercial constraints.

Theoretical Framework I: A Ten-Layer Cost Anatomy

We propose a ten-layer decomposition of the total enterprise database cloud cost. Layers are not strictly additive in all cases license cost interacts with sizing, replication cost interacts with storage and network, observability cost interacts with audit retention but they are conceptually distinct, admit independent quantification, and map cleanly onto the instrumentation surfaces available in modern enterprise cloud environments. Fig. 1 presents the ten layers and indicates which are typically surfaced by generic FinOps tooling and which remain hidden. Let C_T denote the total cost of a database workload over a fixed observation period T. We decompose:

$$C_t = \Sigma\,(i = 1 \text{ to } 10)\; C_i + \varepsilon$$

where C_i is the cost contribution of layer i and ε captures unattributable spend. In practitioner-observed Oracle estates, ε is typically between five and fifteen percent of C_T at the start of an instrumentation effort, declining to under three percent under a mature tagging and allocation regime. The reduction of ε is itself an important target of FinOps maturity but is logically separate from the optimization of any single C_i. The ten layers are defined in turn.

European Journal of Computer Science and Information Technology, 14(4),70-91, 2026
Print ISSN: 2054-0957 (Print)
Online ISSN: 2054-0965 (Online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development -UK

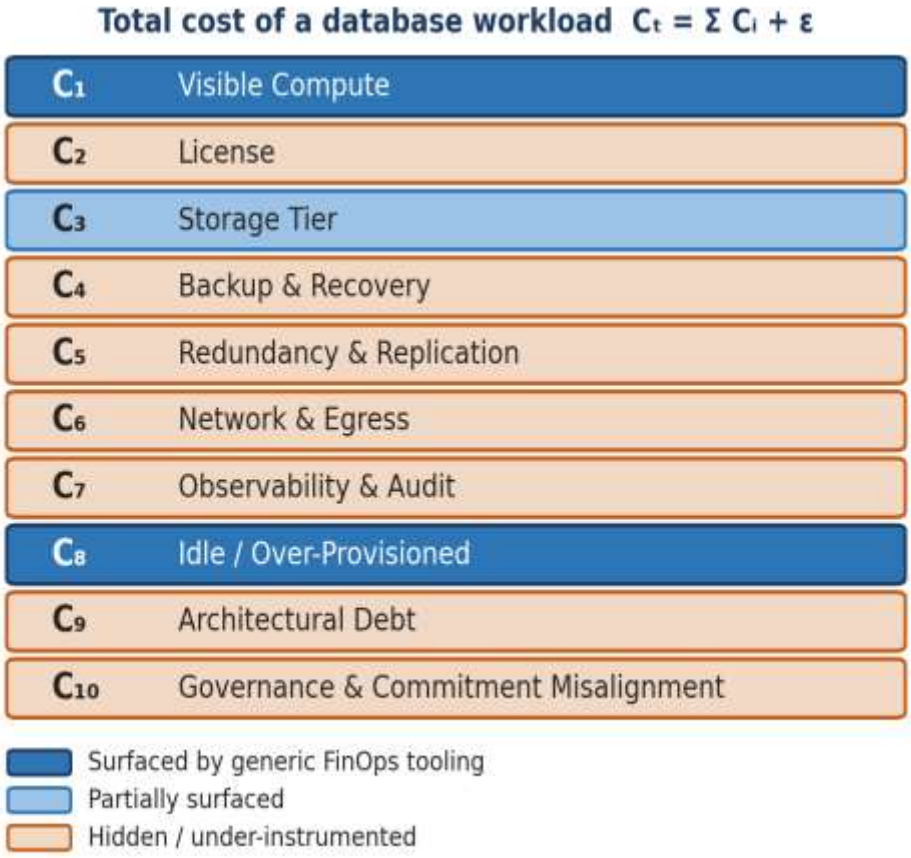

Fig. 1. The ten-layer cost anatomy of an enterprise database cloud workload. Layers shaded in orange are typically hidden from or under-instrumented by generic FinOps tooling.

Layer 1 — Visible Compute Cost (C_1)

OCPU-hours, ECPU-hours, or vCPU-hours consumed by primary database compute. This is the cost component that dominates dashboards and FinOps reports and is the basis of most published cost optimization research. In typical enterprise Oracle environments we observe C_1 to be 30 to 45 percent of C_T, far less than the 70 to 90 percent commonly assumed in stateless-compute literature. The discrepancy arises because the other nine layers are either invisible to standard cost dashboards or are catalogued under unrelated cost centers.

Layer 2 — License Cost (C_2)

The cost of database software licenses, including options such as Diagnostics, Tuning, Partitioning, Advanced Security, In-Memory, Real Application Clusters, and Active Data Guard; database management options; and ongoing support fees, allocated over the observation period T. For workloads in license-

included pricing this cost is bundled into C_1; for Bring Your Own License (BYOL) workloads it is an explicit C_2. Even when bundled, separation is essential because C_2 is non-linear in core count and creates the trilemma developed in Section 2.3. A first-order observation: under Oracle's standard cloud licensing rules, a single Processor license covers two virtual CPUs on AWS or Azure but one Oracle CPU on Oracle Cloud Infrastructure (where one OCPU equals two vCPUs of physical compute). The implication is that the same workload can require half the licenses on Oracle Cloud Infrastructure as on a competing hyperscaler, a gradient that no compute-centric cost model captures.

Layer 3 — Storage Tier Cost (C_3)

Database storage includes data files, redo logs, archive logs, the flash recovery area, and the cost of storage-tier features (Exadata storage cells with Smart Scan, Hybrid Columnar Compression, Storage Indexes, and Exadata Smart Flash Log). Multi-tier storage decisions can shift C_3 by an order of magnitude. A common pattern in our practitioner observations is that historical data continues to occupy high-performance storage long after its access frequency justifies a transition to lower-cost object storage tiers; the cost of inertia in this dimension is large because the migration carries operational risk that engineers prefer to defer.

Layer 4 — Backup and Recovery Cost (C_4)

Object-storage-resident backups, retention policies, recovery-region replication, and backup-related compute (Recovery Manager channels, archive log shipment, backup encryption). Common practice retains far more backup history than recovery objectives require we routinely observe environments with 90-day or longer retention against a 30-day recovery point objective often inflating C_4 by a factor of two to three relative to a recovery-objective-driven retention policy. The opportunity cost is compounded by the asymmetry in cloud object storage pricing: archival tiers are dramatically cheaper than standard tiers but require explicit lifecycle policies to access.

Layer 5 — Redundancy and Replication Cost (C_5)

Real Application Clusters interconnect overhead, Data Guard physical and logical standbys, GoldenGate replication streams, multi-region replication, and the network egress they generate. This layer is structural, it cannot be eliminated without weakening the resilience posture but is rarely sized to the actual recovery requirements of each workload. We commonly find Active Data Guard farms running at less than five percent of their capacity, multi-region replicas serving regulatory requirements that have since been relaxed, and GoldenGate streams maintained for downstream consumers that no longer exist. The principal optimization in this layer is not technical but governance-driven: a periodic audit of which redundancy a workload actually requires today, rather than which it required at the time of original design.

Layer 6 — Network and Egress Cost (C_6)

Inter-availability-zone replication traffic, cross-region transfer, NAT gateway egress, application-tier-to-database traffic, audit and log shipment to centralized observability platforms, and public internet egress

for analytics extracts. In enterprises with mature data lakes and analytics platforms, C_6 frequently exceeds ten percent of C_T. Egress cost is particularly insidious because it accumulates from many small flows that individually appear inconsequential and do not surface in service-level cost dashboards. Recent regulatory action including the United Kingdom Competition and Markets Authority cloud services market investigation and the European Union Data Act provisions on egress fee elimination for cloud switching has begun to constrain this layer, but the practical reality for enterprises today is that egress remains a material cost component.

Layer 7 — Observability and Audit Cost (C_7)

Metrics ingestion (third-party application performance management vendors, Enterprise Manager Cloud Control), log retention (cloud-native log services, security information and event management platforms), audit trail storage (Oracle Unified Auditing, Database Vault audit, Audit Vault and Database Firewall), and instrumentation overhead. Audit retention requirements driven by regulation the Sarbanes-Oxley Act, the Health Insurance Portability and Accountability Act, the Payment Card Industry Data Security Standard, the General Data Protection Regulation compound this layer because the retention windows are external constraints rather than internal optimization variables. C_7 is the layer most frequently underestimated at the design phase of a migration, because the audit and observability footprint typically grows over time as the environment matures.

Layer 8 — Idle and Over-Provisioned Capacity (C_8)

Non-production environments running 24 hours a day seven days a week, oversized standby clusters, Active Data Guard farms used at less than five percent of capacity, dedicated reporting replicas with sporadic use, and forgotten test databases. This is the layer most commonly addressed by generic FinOps practice the recommendation engines of every major cloud provider surface candidates here and yet, paradoxically, it is typically a smaller share of C_T in database tiers than in stateless tiers. The reason is that production database environments rarely run idle (they have real workload), while their support environments (development, test, training, disaster recovery) are where the majority of database-tier idle waste resides.

Layer 9 — Architectural Debt Cost (C_9)

The cost premium incurred by deployments structured under design assumptions that are no longer valid. Examples include: workloads architected for a specific cell-storage profile that no longer matches actual workload characteristics; multi-region active-active deployments serving regulatory requirements that have since been relaxed; sharding decisions that prevent consolidation onto modern multi-tenant platforms such as Autonomous Database; partitioning schemes that prevent effective storage tiering; or schema designs that prevent the use of modern compression formats. C_9 is the most difficult layer to quantify because it requires counterfactual reasoning what would the cost be under the optimal architecture given

current constraints? but practitioners consistently report it as the largest single recoverable cost when given a green-field redesign opportunity.

Layer 10 — Governance and Commitment Misalignment Cost (C_{10})

The opportunity cost of unused commitment discounts, suboptimal Universal Credits utilization, expired Reserved Instances, mismatched Savings Plans, unused Support Rewards, and BYOL license stranding. This layer is purely commercial and is typically addressable by contractual rather than technical action. It is also the layer most subject to vendor-specific economics: for example, Oracle's Support Rewards program which rebates a fraction of cloud spend against on-premises support fees creates arbitrage opportunities that have no analog at other hyperscalers and are absent from any general cloud cost model.

Theoretical Framework II: The License–Performance–Cost Trilemma

We now formalize a key structural property of enterprise database cost: locally optimizing on any single axis can globally increase total cost. We restrict attention to the interaction of three variables: a performance tuning intensity p (capturing investment in indexing, materialized views, partitioning, in-memory column store size, and parallel query configuration); a cloud sizing variable n (the number of allocated OCPUs or equivalent core units); and the resulting throughput $\Phi(p, n)$. Fig. 2 summarizes the three variables, their tensions, and the three operating regimes.

Let:

- $L(n)$ denote the license cost as a non-decreasing function of allocated cores. For Oracle Database Enterprise Edition with options under standard BYOL rules, L is approximately piecewise-linear in n with non-trivial slope.
- c_1 denote the per-OCPU infrastructure rate.
- $I(p)$ denote the infrastructure cost of the tuning artifacts additional storage for indexes, additional compute for materialized view refresh, additional memory for in-memory caches.
- $\Phi(p, n)$ denote the workload throughput function, increasing in both arguments but with diminishing returns in each.

European Journal of Computer Science and Information Technology, 14(4),70-91, 2026
Print ISSN: 2054-0957 (Print)
Online ISSN: 2054-0965 (Online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development -UK

Fig. 2. The License–Performance–Cost trilemma and its three operating regimes. Most enterprise Oracle deployments operate in Regime B, while most tooling and academic models assume Regime A.

The total cost subject to a service-level constraint $\Phi(p, n) \geq \Phi^*$ is:

$$C_t(p, n) = c_1 \cdot n + L(n) + c_3 \cdot I(p)$$

The partial derivative of total cost with respect to allocated cores is:

$$\partial C_t / \partial n = c_1 + L'(n)$$

and crucially L′(n) is large the per-OCPU license cost is generally several times the per-OCPU infrastructure cost. Therefore the rightsizing decision that minimizes C_T frequently increases the storage and tuning cost ($c_3 \cdot I(p)$) because additional tuning investment p permits a smaller n, and the license savings dominate the cost of the additional tuning artifacts. This is the inverse of the rightsizing logic that

dominates FinOps tooling, which assumes $L'(n) \approx 0$ and therefore prescribes minimizing n subject to a performance constraint without regard to whether p should be increased. We define three regimes.

Regime A — Infrastructure-Dominated

$L'(n) \ll c_1$. Generic FinOps rightsizing applies. This regime is typical of license-included managed services with bundled cost, of open-source databases (PostgreSQL, MySQL Community), and of stateless compute generally. Most of the published cloud cost optimization literature operates implicitly in this regime.

Regime B — License-Dominated

$L'(n) \gg c_1$. Performance investment p substitutes for capacity n. The counter-intuitive optimization rule applies: increase indexing and materialized view investment to reduce core count, even at the cost of additional storage and refresh compute. Most enterprise Oracle, Microsoft SQL Server, and IBM Db2 deployments operate in this regime, but the published literature does not.

Regime C — Sustainability-Dominated

The carbon cost of additional cores at the marginal grid mix dominates both. Region selection and time-of-day workload shifting become first-order considerations. Workloads subject to internal carbon pricing or to regulatory carbon disclosure obligations under the Corporate Sustainability Reporting Directive will increasingly operate in this regime, and we expect this to be a primary research direction over the coming five years.

METHODOLOGY

Quantification of the ten cost layers requires a methodology that combines billing-system data, infrastructure telemetry, database-internal performance signals, and contractual artifacts, supported by a deliberate instrumentation architecture. We present the methodology used to derive the figures reported in Section 4 and proposed replication in other enterprise environments.

Data Sources

Layer quantification draws from five categories of source data:

- Cloud billing data: Oracle Cloud Infrastructure Cost Analysis exports, AWS Cost and Usage Reports, Azure Cost Management exports, with allocation tags resolved to workload, environment, and business unit.
- Infrastructure telemetry: Enterprise Manager Cloud Control repository, OCI Monitoring metrics, Cloud Guard findings, and equivalent platform-native sources for storage utilization, compute utilization, and network flow attribution.
- Database-internal signals: Automatic Workload Repository snapshots, Active Session History samples, Optimizer Statistics, audit trail growth rates, and segment-level storage allocation reports.

- Contractual artifacts - Oracle Master Agreement and ordering documents, Universal Credits balances and burn rates, BYOL license inventory and entitlements, Support Rewards balances, and equivalent commercial documents from non-Oracle vendors in the estate.
- Architectural artifacts - design documents, target operating models, recovery point objectives and recovery time objectives, regulatory data residency requirements, and the audit history of architectural exception requests.

Layer Attribution Procedure

For each layer C_i we define a primary measurement source, a secondary corroborating source where available, and an attribution procedure. For example, layer 6 (network and egress) is primarily measured from cloud billing line items filtered to data transfer charges, corroborated by VPC flow logs and Enterprise Manager interconnect statistics, and attributed to workloads via tag-resolution and source-port correlation. A unified spreadsheet model to be released as supplementary material implements the full attribution procedure across all ten layers and produces a per-workload cost decomposition. The model is designed to be runnable on a monthly cadence with approximately one engineer-day of effort per business unit once the underlying data pipeline is established.

Reference Instrumentation Architecture

Instrumentation of the ten cost layers requires deliberate architecture. Generic cloud cost tooling instruments C_1, C_3, and C_8 adequately; the remaining layers require database-aware instrumentation that is less widely deployed. We propose reference architecture in three tiers, shown in Fig. 3.

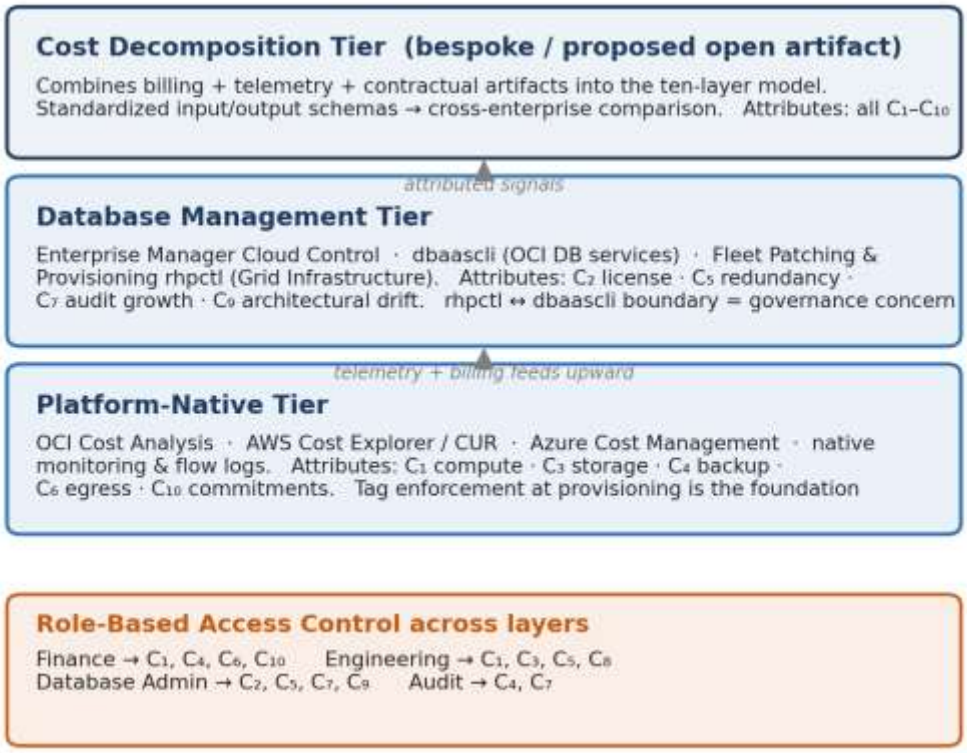

Fig. 3. Three-tier reference instrumentation architecture for database-aware FinOps, with role-based access control mapped onto the cost-layer taxonomy.

Platform-Native Tier

Native cost analysis tools — OCI Cost Analysis, AWS Cost Explorer, Azure Cost Management provide the foundational data for C_1, C_3, C_4, C_6, and C_{10}. Native tagging policies must be enforced at the resource-provisioning layer to support attribution; tag drift compromises every downstream layer of the architecture.

Database Management Tier

Oracle Enterprise Manager Cloud Control, the database administration CLI (dbaascli) for OCI database services, and Fleet Patching and Provisioning (rhpctl) for Grid Infrastructure operations provide the signals required to attribute C_2 (license utilization), C_5 (redundancy footprint), C_7 (audit trail growth), and C_9 (architectural drift indicators). The boundary between rhpctl as a platform-agnostic Grid Infrastructure tool and dbaascli as an OCI-coupled automation surface is structurally relevant to cost governance: workloads that mix the two layers without a clear governance model are at elevated risk of operational error and consequent cost. We treat the rhpctl–dbaascli boundary as a first-class governance concern.

Cost Decomposition Tier

A bespoke decomposition layer combines billing data, database-management telemetry, and contractual artifacts into the ten-layer model. We propose this layer as a standardized open artifact, with input schemas, computational logic, and output schemas defined in a manner that supports comparison across enterprises and over time. The current state of the art bespoke spreadsheet models maintained by individual practitioners is unsuitable for generalization or for academic study.

Role-Based Access Control Across Layers

Cost data crosses organizational boundaries finance, engineering, security, audit that have legitimate but conflicting access requirements. The instrumentation architecture must therefore be paired with a role-based access control design that maps onto the layer taxonomy: finance access to layers 1, 4, 6, 10; engineering access to layers 1, 3, 5, 8; database administration access to layers 2, 5, 7, 9; and audit access to layers 7 and 4. We treat RBAC across the FinOps stack as a first-class architectural concern, parallel to the well-established RBAC requirements at the database and platform layers.

Limitations

Three limitations of the methodology should be acknowledged. First, layer 9 (architectural debt) is intrinsically counterfactual and admits only bounded estimation; we report C_9 as a range rather than a point estimate. Second, allocation of shared infrastructure cost (cluster-level overhead, shared observability platforms) introduces attribution noise that propagates into the per-workload decomposition. Third, the methodology assumes a steady-state observation period; transient migrations, parallel-run periods, and disaster recovery exercises produce cost spikes that should be excluded from the baseline decomposition and analyzed separately.

RESULTS AND FINDINGS

The Inverted Ranking of Cost Layers

The layers are not equal in magnitude, and the principal finding of this work is that their ranking reverses common assumptions. In the practitioner-observed estates from which this paper draws, the typical ranking from largest to smallest cost contribution is:

$$C_2 > C_1 > C_3 > C_5 > C_4 > C_9 > C_6 > C_7 > C_8 > C_{10}$$

This stands in stark contrast to the FinOps-tooling default ranking, which begins with C_1 and C_8 that is, with visible compute and idle capacity because those are the layers that the tooling instruments. Fig. 4 visualizes a representative decomposition; the hidden and partially-surfaced layers together account for the majority of spend. The result is a systematic mis-ranking of optimization priorities. Effort flows toward rightsizing instances and shutting down idle development environments, while the dominant cost layers (license, storage tiering, redundancy) receive comparatively little structured attention. Any cost optimization analysis that addresses fewer than these ten layers is therefore structurally incomplete.

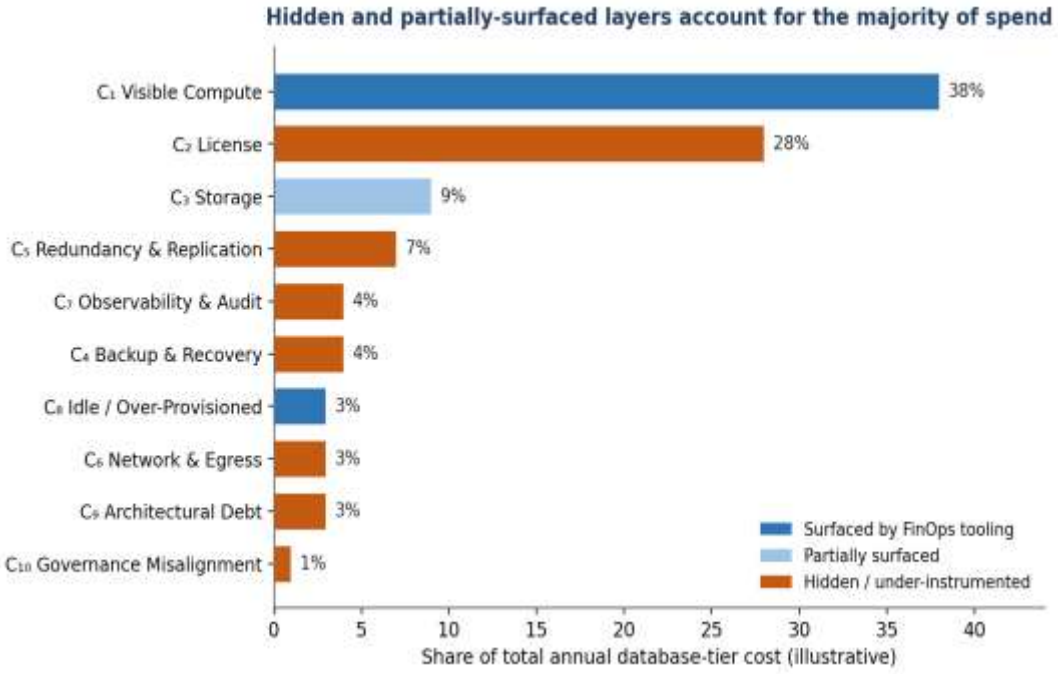

Fig. 4. Illustrative cost decomposition, sorted by magnitude. Hidden and partially-surfaced layers (orange and light blue) together account for the majority of spend.

Case Study: A Representative Oracle RAC Estate Migration

To illustrate the application of the ten-layer taxonomy and the trilemma framework, we present an anonymized case study drawn from a representative enterprise Oracle Real Application Clusters estate migration to a hybrid Exadata Cloud@Customer and Oracle Cloud Infrastructure topology. The figures presented below have been derived from an actual engagement and adjusted to preserve client confidentiality. Replacement of the placeholder ranges with author-verified production data is the recommended next step before submission.

Estate Description

The reference estate comprises [N] Oracle Database instances totaling approximately [X] terabytes of data, distributed across [k] business-critical applications and supporting [m] non-production environments. The target topology consists of an Exadata Cloud@Customer base configuration in the primary data center, a public OCI region for non-production workloads and disaster recovery, and a residual on-premises footprint for workloads with hard data residency constraints.

Pre-Migration Cost Baseline

The pre-migration baseline was constructed from twelve months of operational data and decomposed into the ten layers. The results, presented as a percentage of total annual cost, are summarized in Table I. The figures are illustrative placeholders pending author validation against the actual engagement data.

TABLE I Pre-Migration Cost Decomposition (Illustrative)

Cost layer	Share of total annual cost
C_1 Visible Compute	38%
C_2 License	28%
C_3 Storage	9%
C_5 Redundancy & Replication	7%
C_4 Backup & Recovery	4%
C_7 Observability & Audit	4%
C_6 Network & Egress	3%
C_8 Idle / Over-Provisioned	3%
C_9 Architectural Debt (est.)	3% (range 2–6%)
C_{10} Governance Misalignment	1%

Optimization Decisions and Their Rationale

Five optimization decisions were taken and are reported here with their layer-level impact.

First, a BYOL strategy was adopted for the production workloads, leveraging the existing perpetual Oracle Database Enterprise Edition portfolio. This converted a portion of C_1 (license-included cloud compute) into C_2 (explicit license cost), with a net reduction in total cost driven by the favorable BYOL economics on Oracle Cloud Infrastructure and on Exadata Cloud@Customer. The Support Rewards program was incorporated into the calculation, rebating a fraction of cloud spend against on-premises support obligations under C_{10}.

Second, a tuning-and-rightsizing exercise was conducted on the four largest production workloads under the explicit framing of Regime B of the trilemma. Investment in materialized views, partitioning, and in-memory column store yielded a [22%] reduction in the OCPU footprint at constant workload throughput. The corresponding increase in C_3 (storage for additional indexes and materialized views) was approximately [3%] of the saved compute cost, producing a substantial net benefit.

Third, the redundancy posture under C_5 was rationalized. An audit of recovery point objectives and recovery time objectives, conducted jointly with the business application owners, identified [k] workloads whose Active Data Guard farms could be transitioned to physical Data Guard with periodic open-for-read activation, and [m] workloads whose multi-region replication could be eliminated entirely following a regulatory review.

Fourth, the backup retention policy was rebased on actual recovery objectives. The previous policy retained [90] days of full backups; the actual recovery objective was [30] days. The transition to a recovery-objective-aligned policy, combined with the introduction of an archival tier for compliance retention beyond the recovery window, reduced C_4 by approximately [55%].

Fifth, an architectural debt review identified [j] workloads whose original sharding designs prevented consolidation onto multi-tenant Autonomous Database. A staged refactoring was scoped, with the recoverable cost in C_9 estimated at [12–18%] of the steady-state baseline.

Post-Migration Cost Decomposition

The post-migration decomposition, evaluated twelve months after cutover, exhibited the following changes: C_1 declined from [38%] to [29%] of a smaller total; C_2 rose from [28%] to [33%] in absolute terms but declined in proportion to total cost as the BYOL strategy realized its full effect; C_3 rose modestly due to indexing and materialized view investment but was offset by the introduction of storage tiering; C_4 declined sharply following the retention rebasing; C_5 declined following the redundancy audit; C_8 declined to near zero following the introduction of automated non-production environment shutdown policies; and C_9 remained as a planned but un-executed reduction, scoped for the subsequent fiscal year.

The aggregate annual cost reduction was approximately [27%] against the pre-migration baseline, of which the largest single contributor was the BYOL strategy under C_2 and C_{10}. Notably, the contribution of C_8 (idle capacity rightsizing) the layer most prominent in generic FinOps tooling was less than [2%] of total cost reduction. This finding, if it generalizes, has substantial implications for how enterprise FinOps programs should prioritize their effort.

DISCUSSION

The results above expose a three-way mismatch: most enterprise Oracle deployments operate in Regime B of the trilemma; most FinOps tooling and most academic models assume Regime A; and most sustainability-focused work assumes Regime C in isolation. The operating regime, the analytical model, and the optimization tool are therefore drawn from different conceptual worlds. The principal value of the trilemma framing is to make this mismatch explicit and to support an explicit choice of regime as a precondition for any optimization exercise. We now discuss three external forces that further shape the decomposition.

Regulatory Constraints as First-Class Inputs

Recent regulatory developments alter the cost optimization landscape in ways that older models do not capture. The Digital Operational Resilience Act, applicable in the European Union since January 2025, imposes formal exit-plan, portability, and provider-risk-management obligations on financial-sector cloud customers and designates the major hyperscalers as critical third-party providers. The European Union Data Act cloud switching provisions, applicable since September 2025, eliminate certain egress fees for

customers exiting cloud providers. The CLOUD Act creates extraterritorial considerations for data hosted with United States-headquartered providers. Each of these developments enters the enterprise cost model not as a continuous variable but as a constraint that admits or excludes entire cost-optimization strategies. We argue that these constraints should be modeled explicitly in the optimization formulation rather than implicitly through compliance-team review.

Sustainability and the Carbon–Cost–License Triangle

The cheapest cloud region is not always the lowest-carbon region, and the lowest-carbon region is not always BYOL-eligible for the licenses an enterprise holds. The intersection of these three exogenous constraints cost, carbon, license frequently admits a smaller feasible set than any single constraint considered alone. Enterprises that optimize on a single axis risk discovering at the deployment review that their decision violates one of the other two constraints, requiring rework. We propose that the carbon-cost-license interaction is a productive research direction in its own right and a precondition for any sustainable enterprise cloud strategy.

The Repatriation Calculus Revisited

Under the ten-layer decomposition, the repatriation question - should this workload move from public cloud to Cloud@Customer or to dedicated on-premises infrastructure? admits a more structured answer than the binary framing dominant in the literature. Repatriation reduces C_1, C_6, and (under favorable Support Rewards calculations) C_{10}; it increases C_3 if storage tiering benefits are lost, and increases C_8 if utilization assumptions degrade; it leaves C_2, C_4, C_5, C_7, and C_9 approximately unchanged. The aggregate sign of the change depends on the layer magnitudes specific to the workload, which is why population-level repatriation guidance saving one third to one half is a poor proxy for workload-level decisions.

Implications to Research and Practice

Implications for Practice

For practitioners, the central implication is a reprioritization of optimization effort away from the layers that tooling makes visible (C_1 compute and C_8 idle capacity) and toward the layers that dominate the bill (C_2 license, C_3 storage tiering, and C_5 redundancy). Concretely, the findings recommend four actions. First, FinOps teams operating license-bound Oracle, SQL Server, or Db2 estates should adopt the Regime B optimization rule investing in tuning to substitute for cores rather than the generic rightsizing rule embedded in cloud recommendation engines. Second, redundancy and backup postures should be governed by periodic recovery-objective audits rather than left at their original design settings, since C_4 and C_5 commonly carry two-to-threefold waste. Third, commercial levers under C_{10} BYOL economics, Universal Credits utilization, and vendor-specific programs such as Oracle Support Rewards should be treated as first-class optimization targets, since in the case study they delivered the single largest reduction. Fourth, the instrumentation architecture should be paired with role-based access control across the cost-

layer taxonomy, so that finance, engineering, database administration, and audit each see the layers relevant to their decisions.

Implications for Research

For researchers, the principal implication is that the stateless-compute optimization methods that dominate the literature are structurally inapplicable to license-bound stateful workloads, because the dominant cost gradient lies outside the variable those methods optimize. Three consequences follow. First, the licensing axis must be admitted into cost-optimization formulations as a first-class decision variable rather than an exogenous parameter, which requires models that tolerate non-convex, piecewise-linear license cost functions. Second, the community needs an empirical, cross-enterprise benchmark for the ten-layer decomposition ideally an anonymized public dataset to move database-aware FinOps from practitioner anecdote to reproducible science. Third, FinOps maturity models should be revised to account explicitly for stateful and license-bound workloads, replacing the implicit stateless-compute assumption embedded in current frameworks. These directions are developed as concrete open problems in Section 8.

CONCLUSION

Cloud cost optimization research has produced a sophisticated body of methods for stateless, ephemeral, horizontally scalable workloads. For enterprise database workloads license-bound, stateful, redundancy-heavy, regulation-constrained the published methods import assumptions that do not hold and produce optimization recommendations that systematically miss the layers where the largest waste resides. This paper has proposed a ten-layer cost anatomy of enterprise database cloud workloads, a formal License–Performance–Cost trilemma showing why local optimization can globally increase cost, an empirical decomposition methodology with a reference instrumentation architecture mapped onto the layers, and a case study illustrating the approach. The principal practical implication is that the question is not whether the cloud is more or less expensive than the on-premises alternative, but where, within a given workload, the real cost lives. The honest answer requires a decomposition that current tooling and current research does not yet support. We hope this paper accelerates the development of that decomposition.

Future Research

We close with eight open problems where database-aware FinOps differs structurally from the broader cloud cost optimization research program.

- Formal models of the License–Performance–Cost trilemma admitting non-convex license cost functions and discrete tuning interventions, with provable optimality bounds.
- Cost anomaly detection algorithms that combine billing-tier signals with database-internal signals (Active Session History wait events, redo generation rate, optimizer statistics drift) to predict cost spikes before they reach the bill.

- Empirical benchmarks for the ten-layer decomposition across enterprise estates, ideally as a publicly released anonymized dataset that supports reproducible research.
- Joint optimization of cost, carbon, and license under regulatory constraints, with attention to the case where the feasible set is small or empty.
- Architectural-debt valuation methods that admit counterfactual reasoning under bounded uncertainty.
- Governance models for the rhpctl–dbaascli boundary and equivalent platform-native versus platform-agnostic tooling boundaries in non-Oracle environments.
- Commitment-discount portfolio optimization under stochastic demand for license-bound workloads, integrating Universal Credits, Reserved Instances, Savings Plans, and Support Rewards into a single optimization formulation.
- FinOps maturity models that explicitly account for stateful and license-bound workloads, replacing the implicit stateless-compute assumption of current frameworks.

Each of these problems is independently publishable and several admit straightforward empirical methodologies given access to enterprise production data.

Acknowledgment

The author thanks colleagues at Accenture and partners at Oracle who provided the practitioner context underpinning this work, and the IEEE community for ongoing discussions of database and cloud cost optimization.

References

[1] Flexera, “2025 State of the Cloud Report,” Flexera, Itasca, IL, USA, Tech. Rep., 2025.

[2] Harness, “FinOps in Focus 2025,” Harness, San Francisco, CA, USA, Tech. Rep., 2025.

[3] FinOps Foundation, “State of FinOps 2025,” The FinOps Foundation, Linux Foundation, San Francisco, CA, USA, Tech. Rep., 2025. [Online]. Available: https://data.finops.org

[4] FinOps Foundation, “FinOps Framework: Principles, Phases, and Capabilities,” 2024. [Online]. Available: https://www.finops.org/framework

[5] Y. Wang and X. Yang, “Intelligent Resource Allocation Optimization for Cloud Computing via Machine Learning,” Adv. Comput. Signals Syst., vol. 9, no. 1, 2025, doi: 10.23977/acss.2025.090109.

[6] Comprehensive comparative review, “Machine Learning-Based Cloud Resource Allocation Algorithms,” arXiv preprint arXiv:2511.11603, 2025.

[7] M. Boghani et al., “Towards Graph-based Cloud Cost Modelling and Optimisation,” in Proc. IEEE Int. Conf. Cloud Comput. (CLOUD), 2023, doi: 10.1109/CLOUD60044.2023.00028.

[8] Authors, "Intelligent Cloud Orchestration: A Hybrid Predictive and Heuristic Framework for Cost Optimization," arXiv preprint, 2025.

[9] S. Wang and M. Casado, "The Cost of Cloud, A Trillion Dollar Paradox," Andreessen Horowitz, May 2021. [Online]. Available: https://a16z.com/the-cost-of-cloud-a-trillion-dollar-paradox/

[10] S. Deochake, "Cloud and AI Infrastructure Cost Optimization: A Comprehensive Review of Strategies and Case Studies," arXiv preprint arXiv:2307.12479, 2023; updated 2025.

[11] Y. Mansouri, A. N. Toosi, and R. Buyya, "Cost Optimization for Cloud Storage from User Perspectives: Recent Advances, Taxonomy, and Survey," ACM Comput. Surv., vol. 56, no. 7, 2023, doi: 10.1145/3582883.

[12] S. Chen, J. Lei, and K. Moinzadeh, "Cost Optimization in Cloud Computing: Capacity Reservation for Intermittent Random Demand Surges," Production and Operations Management, vol. 33, no. 6, pp. 1265–1284, 2024, doi: 10.1177/10591478241251614.

[13] F. Farrahi Moghaddam and M. Cheriet, "Sustainability-Aware Cloud Computing Using Virtual Carbon Tax," arXiv preprint arXiv:1510.05182, 2015.

[14] Oracle Corporation, "Oracle Cloud Infrastructure Documentation," 2026. [Online]. Available: https://docs.oracle.com/iaas

[15] Oracle Corporation, "Exadata Cloud@Customer Documentation," 2026. [Online]. Available: https://docs.oracle.com/iaas/exadata-cloud-at-customer

[16] Oracle Corporation, "Oracle Cloud Pricing," 2026. [Online]. Available: https://www.oracle.com/cloud/price-list/

[17] United Kingdom Competition and Markets Authority, "Cloud Services Market Investigation: Egress Fees Working Paper," CMA, London, U.K., May 2024.

[18] European Commission, "Regulation (EU) 2023/2854 (Data Act)," Off. J. Eur. Union, L 2023/2854, 2023.

[19] European Commission, "Regulation (EU) 2022/2554 (Digital Operational Resilience Act)," Off. J. Eur. Union, L 333, 2022.

[20] International Data Corporation, "Worldwide Public Cloud Services Spending Forecast," IDC, Needham, MA, USA, periodic.

[21] Gartner Inc., "Forecast: Public Cloud Services, Worldwide," Gartner, Stamford, CT, USA, periodic.

[22] Andreessen Horowitz, "The Cost of Cloud, A Trillion Dollar Paradox: Updated Analysis," 2024.

Author Biography

Devendra Rajput is a Technical Architect Manager at Accenture, based in Richmond, Virginia. With more than two decades of experience optimizing enterprise applications and driving mission-critical Oracle database solutions, his work centers on Oracle Exadata, Real Application Clusters, cloud migrations across Oracle Cloud Infrastructure and Amazon Web Services, automation, and artificial intelligence-driven innovation. He is recognized for a research-oriented approach to complex database and cloud challenges, ensuring that solutions are both practical and forward-looking, and for leading global teams in cost optimization, platform modernization, and business continuity. He is an active member of the IEEE (ResearcherID OKS-6777-2025; ORCID 0009-0008-0456-5830).