Quantum Cloud Middleware Roll-Ups: API Distribution versus Margin Compression

An M&A Framework for Orchestration Control, Supplier Concentration, Customer Switching Costs and Defensible Value

Authored by Chennakeshav Adya

Independent Researcher

September 2026

Abstract

Quantum cloud middleware connects users, software frameworks, classical compute, simulators and quantum processing units through application programming interfaces, workflow engines and operating controls. A roll-up can widen distribution, consolidate scarce engineering capability and create a common control plane. It can also combine businesses that resell third-party compute, depend on changing provider interfaces and report pass-through usage as revenue. The transaction question is whether the combined company controls a durable customer workflow or remains a thin routing layer exposed to supplier pricing and platform substitution.

This paper develops an M&A and valuation framework for quantum cloud middleware roll-ups. It maps the product from user intent through software development kits, compilation, provider selection, job submission, queue management, classical co-processing, result storage, cost attribution and governance. It tests four sources of value: customer distribution, orchestration and observability, proprietary data and decision logic, and contractual access to compute. It also measures abstraction cost, supplier concentration, provider bypass, service gross margin and the integration burden created by incompatible application programming interfaces.

The evidence base includes current Amazon Braket, Microsoft Azure Quantum and IBM Quantum documentation; open interface and cloud standards; United States merger guidance; financial-reporting requirements; cybersecurity standards; public-company filings; and official cloud-market investigations. Amazon Braket currently exposes devices from several hardware providers and manages regional task execution [1,2]. Azure Quantum can submit common Quantum Intermediate Representation jobs while provider-native targets may retain different formats and parameters [5]. IBM Quantum uses distinct job, batch and session modes with different scheduling and usage consequences [6,7,8]. These facts make middleware commercially relevant while also showing why one universal interface does not remove provider-specific economics or technical behaviour.

A wholly hypothetical target illustrates the valuation method. It has USD 31 million of annual revenue: USD 11 million of orchestration subscriptions, USD 7 million of managed workflows, USD 6 million of professional services and USD 7 million of pass-through compute. Reported gross profit is USD 17.2 million after USD 13.8 million of direct cost. It has 54 paying customers, USD 8 million of opening eligible recurring revenue, USD 9 million of closing eligible recurring revenue, USD 58 million of unrestricted cash and annual cash use of USD 24 million. The two largest compute suppliers represent 72 per cent of QPU spend. Four enterprise-value scenarios produce a probability-weighted value of USD 321.7 million. Every amount, probability, operational measure and scenario is a management assumption created solely to explain the framework.

The analysis concludes that distribution creates value only when the middleware owns a governed customer workflow and earns durable gross profit after compute, cloud, support and scientific delivery

costs. A higher valuation requires portable interfaces, provider-specific optimisation, reliable telemetry, customer decision integration, contractual rights, controlled security and evidence that customers remain after suppliers change. Pass-through compute, one-off integration and promotional credits should be separated from recurring software. Deal terms should pay at closing for controlled software, transferable customer relationships and achieved margin while making platform premiums conditional on retention, supplier diversification, portability and integration milestones.

JEL Classification: G12, G24, G34, L13, L22, L86, O31, O33

Keywords: quantum cloud, middleware, mergers and acquisitions, application programming interfaces, orchestration, supplier concentration, cloud economics, platform valuation, roll-up strategy

Introduction

Quantum computing is delivered through a layered market. Hardware companies operate quantum processing units. Public clouds and hardware vendors expose devices through application programming interfaces. Software frameworks prepare circuits and workloads. Middleware can select providers, compile jobs, coordinate classical resources, track cost, store results and enforce governance. Enterprise users experience the stack as one workflow even though several parties may control its components.

Official documentation demonstrates the commercial importance of this layer. Amazon Braket provides access to multiple gate-based and analogue devices, submits work through regional services and stores results in customer-controlled cloud storage [1,2]. Azure Quantum manages workspaces, targets and job metadata while noting that provider-native jobs can require different formats or parameters [5]. IBM Quantum distinguishes job, batch and session execution because scheduling, exclusivity, latency and budget behaviour differ [6,7]. Middleware can simplify these differences for customers, yet it cannot make the underlying constraints disappear.

Consolidation is therefore plausible. A buyer can combine an orchestration engine, framework adapters, cost management, enterprise security, application libraries and customer access. A roll-up may reduce duplicate engineering and accelerate cross-selling. It may also increase exposure to the same suppliers, combine incompatible architectures and create a larger low-margin reseller. The board needs a method that separates controlled value from aggregated activity.

This paper provides that method. It begins with the customer decision, maps the platform and dependency stack, tests revenue presentation and unit economics, evaluates customer and supplier concentration, and builds a probability-weighted valuation. It then converts uncertainty into transaction terms and an integration plan that preserves portability and customer trust.

1 Define the roll-up thesis as a customer outcome

The acquisition thesis should state the customer problem that the combined business will solve. Examples include giving an enterprise one governed route to several devices, reducing the engineering required to move workloads, improving job reliability, controlling quantum spend, reproducing experiments or integrating quantum tasks into an existing high-performance-computing process. The thesis should identify the accountable user, current alternative, measurable outcome and willingness to pay.

A broad claim that the buyer will create a quantum platform is insufficient. Platform value depends on which interactions the company controls and why users remain. The board should specify whether the desired control point is developer access, workflow orchestration, provider selection, security, data lineage, cost management, application logic or the final business decision. Each position has a different competitive set and valuation basis.

The counterfactual should include direct provider access, public-cloud marketplaces, open-source frameworks, internal engineering and traditional high-performance-computing workflow tools. A

customer may accept a multi-provider abstraction for convenience while retaining the ability to bypass it. The diligence team should test whether the proposed combination changes the customer's cost, speed, risk or governance enough to support recurring payment.

The roll-up thesis should include disconfirming evidence. High application programming interface usage may reflect free trials or promotional credits. A large device catalogue may have limited active use. A unified interface may expose only the lowest common denominator. The board should define the evidence that would cause it to reduce price, change structure or stop the transaction.

2 Map the platform from user intent to result

The platform map should follow one workload end to end. It begins with the user's problem and source code, then covers framework translation, circuit or programme representation, compilation, optimisation, provider and device selection, authentication, job submission, queueing, classical co-processing, execution, error handling, result retrieval, storage, cost allocation and decision reporting. Each step should identify the controlling party and transferable asset.

This map reveals whether middleware is genuinely central. A company may provide an attractive portal while provider software performs compilation and execution. Another may expose one application programming interface but rely on separate adapters and manual support. A third may own policy, scheduling, telemetry and workflow state across environments. The third position can support stronger switching costs if the controls are secure, reliable and accepted by customers.

The buyer should distinguish the control plane from the data plane. The control plane manages identity, policy, routing, versions, jobs, budgets and records. The data plane carries circuits, parameters, results and associated information. Ownership of the control plane can create value because it governs repeated use. It also creates responsibility for security, availability and data rights.

Every hand-off needs an evidence record. The diligence team should capture interface versions, service levels, provider terms, failure modes, retry logic, latency, queue behaviour, data location and support ownership. A diagram without run histories and contracts cannot establish operational control.

3 Identify the customer job and the decision pathway

Middleware is valuable when it supports a customer job that continues despite changes in hardware. The job may be experimentation, algorithm benchmarking, application development, workload scheduling, cost governance or regulated research. The board should identify the point at which the middleware output enters the customer's technical or commercial decision.

For a research team, the relevant outcome may be a reproducible comparison across devices. For an enterprise platform team, it may be controlled access and budget allocation. For an application company, it may be reliable execution of a hybrid workflow. The same software can serve all three, but the evidence and willingness to pay differ.

The diligence team should trace a sample customer from initial setup through repeated use. Required evidence includes user roles, workloads, active providers, successful and failed jobs, support tickets, cost records, result use and renewal. Interviews should confirm which feature would be hardest to replace and whether the customer can move directly to a supplier.

Customer outcomes should be measured after the complete process. A faster submission interface has limited value when queue time, compilation, data preparation or scientific review remains dominant. The value case should quantify engineering hours, failed runs, governance effort, compute cost and decision cycle time before and after adoption.

4 Distinguish orchestration software from compute resale

Quantum middleware can earn subscription fees, usage fees, managed-service fees, professional-service revenue and a margin on third-party compute. These streams should be separated because their economics differ. Subscription revenue can support a software multiple when customers pay for controlled functionality. Compute resale may be pass-through activity whose value depends on contractual spread, working capital and supplier access.

IFRS 15 requires an entity to assess whether it controls a specified good or service before transfer when another party participates in delivery [18]. A principal generally records gross consideration; an agent records its fee or commission. The legal and accounting conclusion depends on contract facts, including responsibility, inventory risk and pricing discretion. Transaction diligence should therefore reconcile reported revenue with the underlying promise and control evidence.

The buyer should calculate gross profit after QPU charges, simulator cost, classical compute, storage, network, support, credits and refunds. Promotional credits should not be treated as sustainable margin. Unused minimum commitments should be included in supplier economics. A reseller can show fast revenue growth while gross profit and cash contribution remain weak.

Orchestration value should be tested independently of resale. The team can price the software without bundled compute, compare direct and indirect routes, and measure renewal among customers whose supplier usage changes. A platform that retains the customer while providers change has stronger evidence of controlled value.

5 Test interface portability and the abstraction tax

Portability has several levels. Source portability means a programme can be expressed in another framework. Build portability means dependencies and environments can be recreated. Execution portability means a workload can run through another provider. Performance portability means it remains efficient. Result portability means outputs and provenance remain usable after migration.

Common representations can reduce friction. Azure Quantum documentation describes submission of Quantum Intermediate Representation jobs and also notes that provider-native jobs may require different formats or parameters [5]. OpenQASM and QIR provide useful interface standards [10,11]. Framework plugins can widen access [12,13]. Standards support translation; they do not guarantee equal native gates, calibration, topology, scheduling, mitigation or price.

The abstraction tax is the difference between the best provider-specific implementation and the middleware route. It can appear as additional latency, lower circuit efficiency, missing features, slower adoption of new capabilities, more support or reduced observability. The buyer should benchmark representative workloads through the middleware and directly through provider tools.

A defensible platform can manage the tax transparently. It may expose a portable core while allowing provider-specific extensions, preserve intermediate representations and record every transformation. Customers can then choose between portability and optimisation with evidence. A lowest-common-denominator interface that hides material differences can increase risk and weaken trust.

6 Measure supplier concentration and bargaining power

The supplier map should identify every hardware provider, public cloud, simulator, classical-compute service and critical software dependency. For each relationship, diligence should record spend, workload share, contract term, pricing, credits, termination, data processing, service levels, feature access, roadmap dependence and replacement path.

Concentration should be measured in several ways. Spend concentration shows economic exposure. Workload concentration shows operational reliance. Customer concentration by supplier reveals whether a provider change threatens particular accounts. Feature concentration shows whether a proprietary capability is difficult to replace. Geographic concentration can create data-residency or service-continuity risk.

Current public evidence shows why the test matters. Amazon Braket lists devices from several hardware providers [2]. IonQ reports availability through major cloud platforms and its own service [23]. Rigetti describes its proprietary cloud service and public or private cloud integration [24]. These routes widen distribution while giving hardware suppliers and clouds direct customer relationships.

The buyer should model supplier responses to a roll-up. A supplier may welcome additional demand, reduce discounts, change interface terms, prioritise its own service or approach customers directly. Contractual protections, technical alternatives and customer ownership determine whether the middleware can maintain margin.

7 Reconstruct unit economics by workload

Unit economics should be built from individual workloads rather than consolidated averages. For each workload class, the buyer should calculate customer price, QPU usage, simulator and classical compute, storage, network, support, scientific labour, credits, failures, refunds and payment cost. The result should show contribution before shared research and corporate overhead.

Execution modes influence economics. Amazon Braket Hybrid Jobs combines classical resources with quantum processing and prioritises job tasks while resources remain active [3]. IBM job, batch and session modes have different scheduling and usage characteristics [7,8,9]. Middleware that chooses the appropriate mode can reduce cost or latency. Poor routing can increase both.

The team should compare quoted and realised margin. Minimum commitments, idle reservations, queue failures and repeated jobs can erode contribution. A customer may receive a fixed price while the platform bears usage volatility. Usage caps, repricing rights and automated budget controls can improve resilience.

Gross margin should be reported for software, managed workflows, services and resale separately. Blended margin can conceal a growing low-margin stream. The valuation model should apply a software multiple only to revenue supported by recurring software economics and customer evidence.

The analysis should also follow margin through scale. Additional workloads can improve software contribution when infrastructure and support remain stable. They can reduce contribution when new devices require bespoke adapters, more scientific support or committed capacity. The board should review incremental gross profit by customer and provider rather than assuming that aggregate usage creates operating leverage. A useful sensitivity table varies supplier price, failure rate, support time and customer price together. This exposes contracts whose apparent growth consumes cash.

Working capital deserves separate treatment. Marketplace settlement cycles, customer prepayments, supplier commitments and refundable credits can create a gap between reported gross profit and cash. The buyer should reconcile monthly billings, provider invoices, collections, deferred revenue and minimum commitments. A roll-up can improve purchasing power, yet the combined entity may inherit several overlapping commitments. Integration planning should quantify cancellation cost and the order in which agreements can be consolidated.

8 Audit pricing, metering and cost attribution

Measured service is a core cloud characteristic under the NIST definition [14]. Quantum middleware should preserve a meter from customer request through every provider charge. The meter should reconcile

jobs, shots, circuits, reservations, classical resources, storage, credits, taxes and refunds to invoices and general-ledger entries.

Pricing can be subscription based, per task, per shot, per minute, per reservation, per workflow or outcome linked. Each model shifts risk differently. A per-workflow price can simplify purchasing while exposing the vendor to provider cost variation. A pass-through model protects margin while reducing differentiation. Enterprise contracts may combine a platform fee with controlled usage.

The buyer should test whether the target can explain a sample invoice. It should reproduce the customer's charge from raw provider records and pricing rules, identify exceptions and show approval. Unreconciled usage creates margin leakage and customer disputes. Missing telemetry also weakens cohort and valuation analysis.

Cost attribution should include failed and cancelled jobs. A failed job may still consume classical resources or support time. The platform should distinguish provider failure, customer error, middleware defect and scientific non-convergence. This information supports supplier claims, product improvement and accurate gross margin.

9 Evaluate telemetry, data rights and the control plane

Operational telemetry can become an important asset. Job histories, provider performance, queue time, failure patterns, compilation choices, cost and customer workflow context can improve routing and support. Value depends on lawful rights, data quality, coverage and demonstrated contribution.

The buyer should classify customer inputs, circuits, parameters, provider metadata, results, support records and aggregated analytics. Contracts should define ownership, confidentiality, permitted processing, retention, deletion, model training and cross-customer use. Technical access does not create a right to reuse confidential workloads.

Routing logic should be explainable and testable. The platform may select a provider based on compatibility, availability, fidelity, cost, geography or customer policy. Diligence should reproduce decisions, identify overrides and measure whether routing improved the intended outcome. Proprietary claims require evidence beyond a rule table that competitors can recreate.

Data lineage should connect the original workload to every transformation and result. A complete record supports reproducibility, audit, security and customer trust. It also reduces integration risk because the buyer can migrate records without losing meaning.

10 Analyse customer cohorts and switching costs

Customer analysis should begin with contracts and cash. The team should identify paying customers, active users, recurring revenue, compute resale, services, credits, deferred revenue and collections. Product telemetry should reconcile to the commercial record.

Cohorts should show opening recurring revenue, expansion, contraction, churn, new recurring revenue and closing recurring revenue. Expansion should be separated into higher software value and additional pass-through usage. A customer whose total bill rises because QPU prices increase has not necessarily adopted more middleware.

Switching-cost evidence includes embedded authentication, policy, workflow definitions, cost controls, result repositories, audit records and application integrations. Migration time should be tested with a representative customer environment. Contract duration alone does not establish product dependence.

Concentration remains important. A small number of research partners or government programmes can dominate early quantum revenue. Rigetti's 2025 filing reported substantial government exposure and

several significant customers [24]. Public-company evidence does not describe a hypothetical target; it illustrates why customer concentration and revenue quality require direct testing.

11 Review contracts, licences and ecosystem rights

The contract review should cover customer subscriptions, provider access, marketplace terms, framework licences, open-source components, data processing, subcontracting, service levels, export controls and change-of-control provisions. The buyer should identify rights that terminate, require consent or change price after acquisition.

Provider application programming interfaces can change. The integration inventory should record version support, deprecation notice, compatibility tests and remediation time. Amazon's documentation history includes device additions, retirements, quota changes and service updates [4]. Middleware must absorb this change without destabilising customers.

Open-source frameworks can accelerate distribution while reducing proprietary control. The buyer should review licence obligations, contributor rights, trademarks, security and the boundary between open and proprietary components. A large community can create value even when the code is available, provided the company owns trusted operations, enterprise features or customer workflows.

Marketplace arrangements should be separated from direct contracts. The cloud may control billing, customer data, discounts and relationship terms. The buyer should verify whether marketplace listings create transferable customers or a revocable distribution channel.

12 Assess security, sovereignty and operational resilience

Quantum workloads can contain confidential algorithms, portfolio problems, molecular structures and infrastructure data. Middleware may hold credentials for several providers and therefore becomes a high-value control point. Security diligence should cover identity, privileged access, secrets, encryption, software supply chain, logging, incident response and data segregation.

NIST zero-trust guidance requires resource-focused controls without implicit trust based on network location [15]. NIST software-development and supply-chain guidance supports controlled builds, dependencies and release practices [26,27]. The acquirer should test these controls against the actual multi-provider architecture.

Data location and third-party processing should be explicit. Amazon Braket documents regional devices and third-party processing [1,2]. Customer contracts and platform configuration should reflect where workloads and results travel. Sovereignty requirements can limit provider choice and create demand for private or national deployments.

Resilience testing should include provider outage, credential compromise, device retirement, failed job, region loss and corrupted result. The platform should preserve workflow state, notify customers, prevent duplicate cost and support an alternate route. Recovery objectives should be grounded in customer commitments.

13 Perform reproducible technical diligence

The technical review should start from controlled repositories and a clean environment. The buyer should build the platform, deploy a test instance, connect approved providers, execute representative workloads and reproduce results. Every manual action should be recorded.

Testing should cover unit, integration, security, compatibility, load, failure and migration behaviour. Provider adapters require contract tests because a successful response does not guarantee semantic equivalence. The team should verify that versions, units, result formats and error states remain correct.

The code review should separate core orchestration, adapters, user interface, enterprise control, telemetry, scientific logic and deployment infrastructure. Proprietary value may reside in one component while the rest is standard engineering. Replacement-cost and income methods should reflect this allocation.

Founder intervention should be measured. If founders repair adapters, interpret provider errors or manage key accounts personally, the platform has transfer risk. Independent execution by the buyer's team is stronger evidence than documentation alone.

14 Design the roll-up integration architecture

The integration plan should preserve customer continuity before consolidating platforms. The buyer should establish a canonical workload and result model, common identity and policy layer, shared telemetry, contract inventory and migration factory. Each acquired product can then map into controlled interfaces.

A forced early rewrite can destroy value. Customers may depend on provider-specific features or embedded application programming interfaces. The integration sequence should stabilise services, instrument usage, reconcile economics and migrate low-risk components before changing customer-facing behaviour.

The canonical model should preserve provider extensions. A portable core can support shared policy and reporting, while extension fields retain native capability. Governance should prevent acquired adapters from silently changing semantics.

Integration economics need a baseline. The board should record duplicate cloud infrastructure, adapter maintenance, support, sales, research and corporate cost. Savings should be reduced for migration expense, retention, contract consent and customer support. Revenue synergies should require identified accounts, products, owners and conversion evidence.

Customer migration should be governed as a product release. Each migration wave should define eligibility, data mapping, interface compatibility, security approval, acceptance tests, rollback and support coverage. The buyer should begin with low-complexity customers and retain the acquired interface until evidence shows that the canonical model preserves required behaviour. Migration success should be measured through retained use, error rate, support effort, gross profit and customer approval.

People integration should follow capability rather than organisational title. Adapter engineering, provider relationships, security operations, customer architecture and scientific support may be concentrated in small teams. The buyer should map critical people to systems and accounts, document succession and stage knowledge transfer. Retention awards should connect to completed transfer, service continuity and customer outcomes. A larger combined headcount does not create platform capability when expertise remains isolated.

15 Examine platform competition and serial-acquisition risk

Middleware can connect users and multiple suppliers, which can create platform characteristics. United States merger guidance examines competition between platforms, on a platform and to displace a platform [16]. It also considers patterns of multiple acquisitions and trends toward consolidation [17]. A roll-up strategy should assess competition and access before signing each transaction.

The buyer should ask whether the combined firm can degrade rival access, favour an affiliated provider, bundle services, restrict portability or acquire a tool that helps customers use several platforms. These issues can arise even when current revenue is small because control over future interfaces and data may matter.

Antitrust diligence should map suppliers, customers, competing middleware, open-source alternatives and adjacent cloud services. It should test market definition under several future states and preserve evidence of customer benefit. Claimed efficiencies should be specific, verifiable and transaction related.

The integration design can reduce risk. Transparent routing, provider neutrality, export tools, documented interfaces and customer choice can support competition and trust. Governance should record conflicts when the platform has an economic interest in a particular supplier.

16 Build income and replacement-cost cases

The income model should forecast software recurring revenue, managed workflows, services and compute resale separately. Drivers should include active customers, users, workflows, provider usage, price, gross margin, retention, support and integration cost. The model should reconcile revenue to cash and deferred amounts.

Software value should reflect durable gross profit. Pass-through compute can support distribution and data while receiving a lower multiple. Services can enable adoption but require labour and may not scale. The buyer should test downside cases for provider price increases, lost discounts, customer bypass and slower quantum adoption.

Replacement cost should estimate the time and cash required to recreate adapters, orchestration, enterprise controls, telemetry, customer integrations, contracts and team capability. Historical research expense is not automatically value. The analysis should exclude failed work that a rational buyer would avoid.

Replacement time can matter when provider access or customer relationships are scarce. The board should record which assets accelerate entry and which require ongoing retention. The replacement case should remain below the cost of developing a superior alternative unless the acquired business brings defensible customers or rights.

17 Construct the hypothetical valuation

The wholly hypothetical target has USD 31 million of annual revenue. Orchestration subscriptions contribute USD 11 million, managed workflows USD 7 million, professional services USD 6 million and pass-through compute USD 7 million. Direct cost is USD 13.8 million, producing reported gross profit of USD 17.2 million. The model treats these values as assumptions rather than observed company data.

The customer base comprises 54 paying organisations. Eligible recurring revenue opens at USD 8 million and closes at USD 9 million after expansion, new recurring revenue, contraction and churn. The five largest customers represent 49 per cent of revenue. The two largest compute suppliers represent 72 per cent of QPU spend. The target holds USD 58 million of unrestricted cash and uses USD 24 million annually.

Four enterprise-value scenarios are used. A compute reseller with limited control is valued at USD 95 million with 30 per cent probability. An orchestration product with credible enterprise controls is valued at USD 240 million with 38 per cent probability. A multi-provider workflow platform with accepted switching costs is valued at USD 515 million with 24 per cent probability. A category control plane is valued at USD 980 million with 8 per cent probability. The weighted enterprise value is USD 321.7 million.

The illustration allocates USD 78 million to software and intellectual property, USD 62 million to customer relationships, USD 43 million to telemetry and operational data, USD 31 million to provider contracts and access, USD 48 million to team and know-how, and USD 59.7 million to platform options. The allocation is a decision tool; an accounting purchase-price allocation requires qualified analysis under applicable standards [19,20,21,22].

18 Structure consideration and integration

Closing consideration should pay for assets that are controlled and transferable. These include source code, documented adapters, enterprise controls, customer contracts, collected cash and data rights. A holdback can address contract consents, security remediation, working-capital leakage and disputed principal-versus-agent accounting.

Contingent consideration can depend on software retention, customer renewal, supplier diversification, portable workload performance and gross-profit conversion. Milestones should be measurable, time bounded and resistant to manipulation. Revenue alone is a weak metric when pass-through compute can raise reported sales while reducing margin.

The first one hundred days should protect service continuity, credentials, customer communication and provider relationships. The buyer should establish a combined dependency register, telemetry baseline, margin bridge and migration plan. Product consolidation should follow evidence from customer workflows.

The board should maintain an integration value ledger. Every initiative should state baseline, target, cost, owner, dependency, timing and realised result. Unachieved value should trigger a product, capital or transaction decision rather than an unsupported platform narrative.

Conclusion

Quantum cloud middleware can solve a real coordination problem. Customers face changing devices, provider-specific interfaces, hybrid classical resources, queues, pricing and governance. A well-designed control plane can reduce this complexity, preserve evidence and make multi-provider use economically manageable.

A roll-up creates defensible value when the combined business owns repeated customer workflows, maintains portable and provider-aware execution, controls telemetry and security, and converts activity into durable gross profit. Distribution alone is insufficient when suppliers can bypass the platform or when revenue mainly passes through to compute providers.

The transaction method should begin with a customer outcome and follow the complete workload. It should reconstruct unit economics, measure supplier and customer concentration, test portability, verify contracts and reproduce the technology. The valuation should separate software, relationships, data, access, team and future options.

Deal structure and integration can then allocate risk. Upfront price rewards achieved control and transferable economics. Holdbacks and contingent consideration address migration, retention, supplier dependence and future platform evidence. This discipline allows a buyer to pursue scale while preserving technical credibility, customer choice and capital efficiency.

Appendix A Workload diligence protocol

Select representative workloads by customer, framework, provider, device and commercial importance. Reproduce each workload from source through result, record every transformation, compare direct and middleware routes, reconcile cost and time, and identify manual intervention. Retain clean-environment evidence and customer acceptance criteria.

The protocol should include success, failure, cancellation, provider outage and migration cases. Results should feed the dependency register, unit-economics model and integration plan.

Appendix B Customer and supplier evidence schedule

Customer evidence should include contracts, invoices, cash, active users, workloads, support, decision use, renewal, migration effort and direct-provider alternatives. Supplier evidence should include terms, spend, credits, commitments, service levels, roadmap access, data processing, termination, change of control and replacement.

Evidence should be reconciled at customer-workload-provider level. Consolidated summaries can conceal pass-through revenue, concentration and negative contribution.

Appendix C Valuation data room

The valuation data room should contain monthly revenue and gross profit by stream, customer cohorts, provider spend, workload telemetry, pricing rules, credits, contracts, cash, deferred revenue, backlog, forecasts and bridge schedules. Technical folders should include repositories, builds, tests, adapter versions, architecture, security, incidents and migration tools.

Every model input should link to an owner and source. Hypothetical scenarios should remain visibly separate from observed performance.

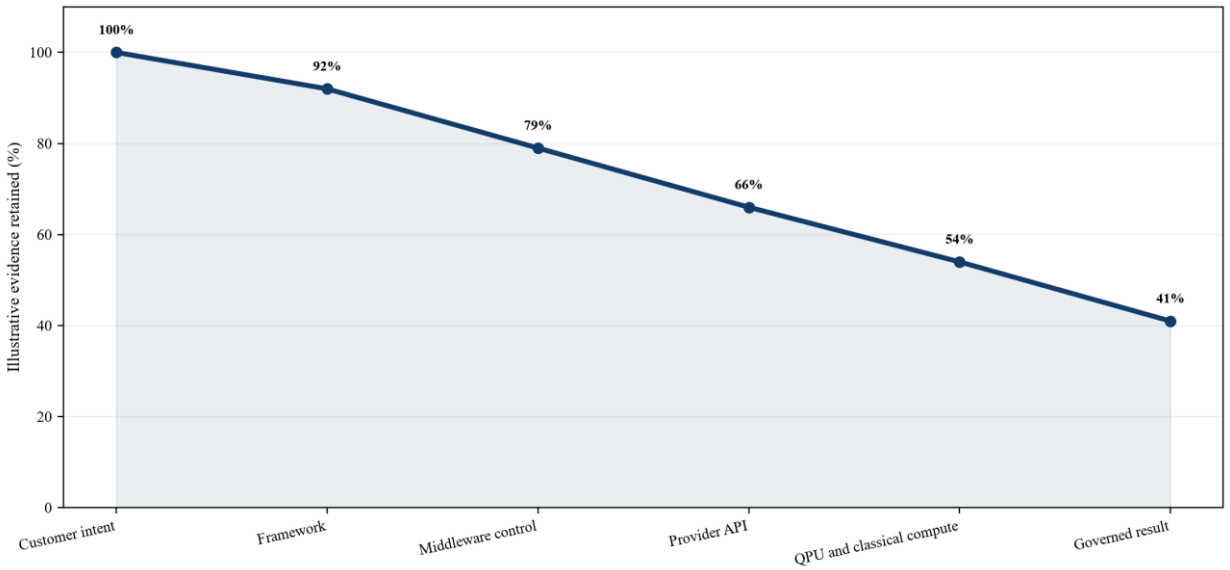
Appendix D Integration value ledger

The ledger should list each value initiative, baseline, target, evidence, owner, cost, timing, dependency and realised result. Initiatives may include provider diversification, adapter consolidation, common identity, telemetry unification, margin improvement, customer migration and security remediation.

The board should review the ledger at defined intervals and preserve a record linking acquisition thesis, operational evidence and cash outcome.

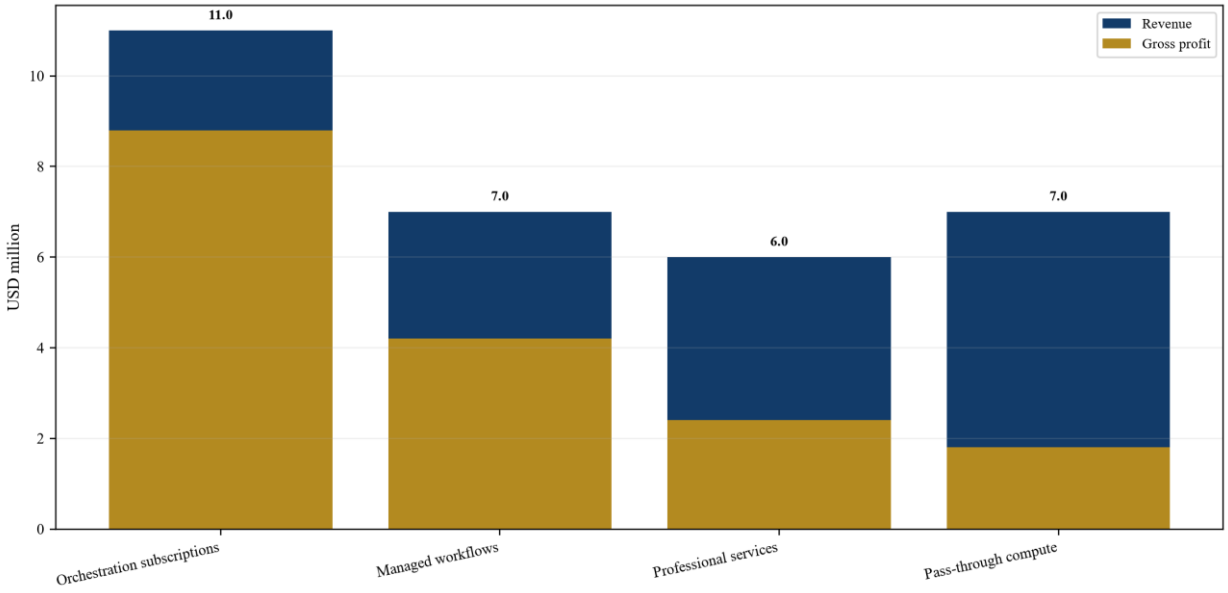
Appendix E Decision figures and tables

Figure 1. Quantum middleware control path from customer intent to governed result



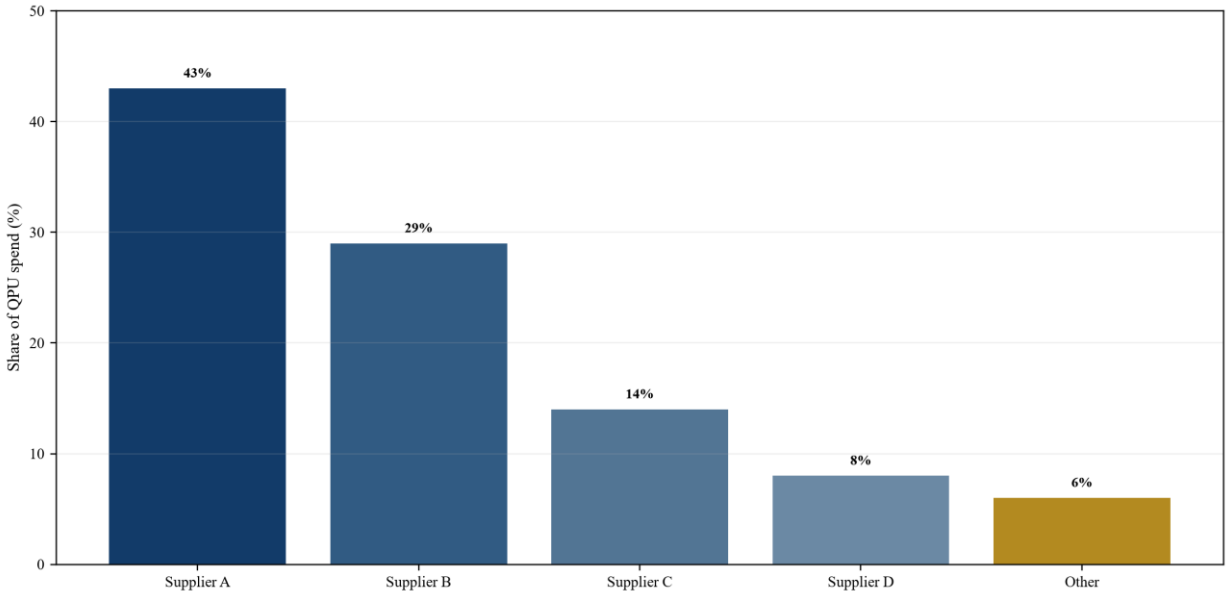
Proposed transaction-diligence architecture; control strength should be evidenced at every hand-off.

Figure 2. Hypothetical revenue and gross-profit bridge



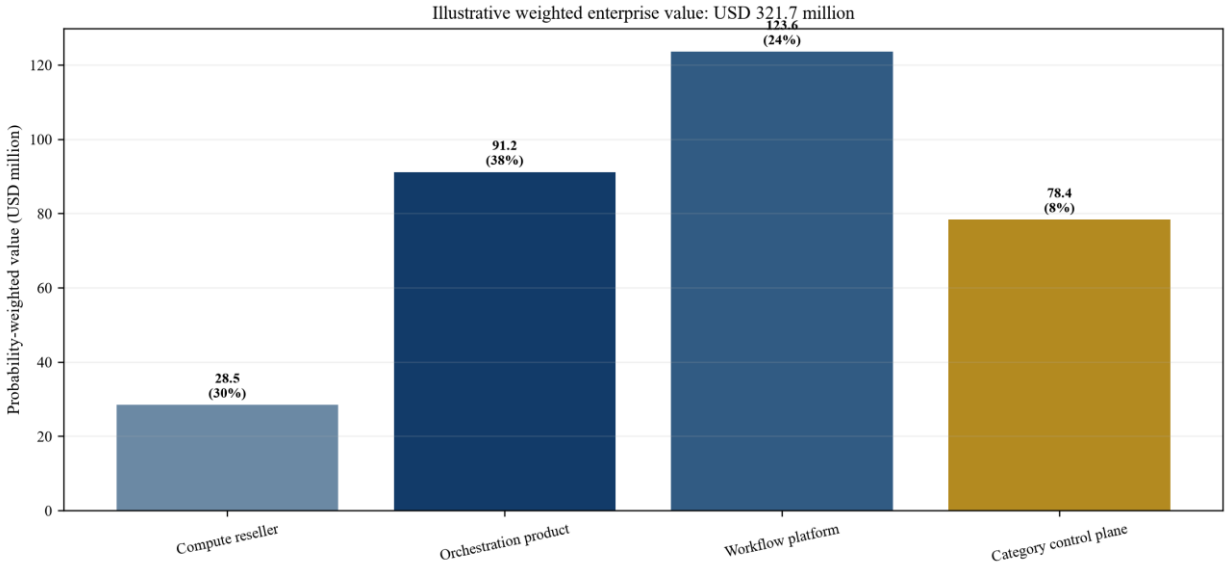
Wholly hypothetical management assumptions; USD million.

Figure 3. Hypothetical supplier concentration by QPU spend



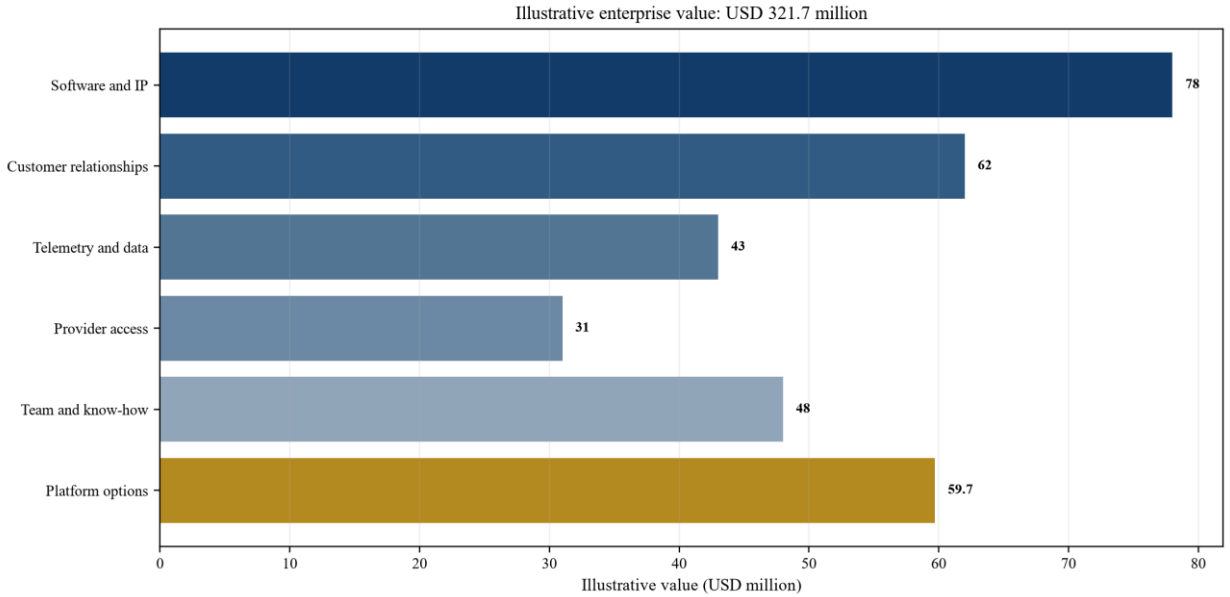
Wholly hypothetical management assumptions; two largest suppliers represent 72 per cent.

Figure 4. Hypothetical probability-weighted enterprise value



Wholly hypothetical management assumptions; USD million.

Figure 5. Hypothetical valuation allocation



Wholly hypothetical management assumptions; USD million.

Table 1. Middleware control matrix

Layer	Control evidence	Principal dependency	Valuation implication
Customer workflow	Repeated governed use	Customer process	Relationship and switching value
Orchestration	Policy, routing and state	Provider interfaces	Software value
Compilation	Reproducible transformations	Framework and target	Technical differentiation
Execution	Jobs, queues and results	Cloud and QPU supplier	Concentration adjustment
Economics	Meter, price and margin	Supplier terms	Income value
Governance	Identity, audit and data lineage	Enterprise controls	Trust and retention value

Proposed diligence classification.

Table 2. Provider-dependency review

Dimension	Evidence	Failure mode	Transaction response
Interface	Version and compatibility tests	Breaking change	Adapter remediation reserve
Economics	Price, credits and commitments	Margin compression	Repricing and diversification
Access	Capacity, queue and service level	Customer disruption	Alternate route
Data	Location, processing and deletion	Contract breach	Policy and consent
Strategy	Roadmap and direct sales	Platform bypass	Customer-control test

Proposed minimum record for every material supplier.

Table 3. Hypothetical recurring-revenue cohort

Cohort component	Illustrative amount	Evidence required	Interpretation
Opening eligible recurring revenue	USD 8.0 million	Contracts and cash	Cohort base
Expansion	USD 1.8 million	Higher software scope	Product growth
New recurring revenue	USD 1.2 million	New paying customers	Acquisition performance
Contraction	USD 0.8 million	Reduced scope	Retention pressure
Churn	USD 1.2 million	Lost contracts	Product or market risk
Closing eligible recurring revenue	USD 9.0 million	Reconciled ledger	Income base

Wholly hypothetical management assumptions; USD million.

Table 4. Hypothetical revenue and unit economics

Revenue stream	Revenue	Gross profit	Review question
Orchestration subscriptions	USD 11.0 million	USD 8.8 million	Is use recurring and independent of resale?
Managed workflows	USD 7.0 million	USD 4.2 million	How much support and scientific labour is required?
Professional services	USD 6.0 million	USD 2.4 million	Can work convert into reusable product?
Pass-through compute	USD 7.0 million	USD 1.8 million	Is gross presentation and spread sustainable?
Total	USD 31.0 million	USD 17.2 million	Does cash reconcile to reported economics?

Wholly hypothetical management assumptions; USD million.

Table 5. Hypothetical valuation allocation

Value component	Illustrative amount	Evidence gate	Downside treatment
Software and intellectual property	USD 78.0 million	Clean build, portability and rights	Reproduction deduction
Customer relationships	USD 62.0 million	Retention, use and cash	Cohort adjustment

Value component	Illustrative amount	Evidence gate	Downside treatment
Telemetry and operational data	USD 43.0 million	Rights, quality and contribution	Rights deduction
Provider contracts and access	USD 31.0 million	Transfer and economics	Concentration reserve
Team and know-how	USD 48.0 million	Independent operation and retention	Service-based retention
Platform options	USD 59.7 million	Supplier diversity and workflow growth	Contingent consideration

Wholly hypothetical management assumptions; not observed company or transaction data.

Table 6. Roll-up integration sequence

Phase	Principal action	Evidence gate	Board output
Stabilise	Protect service, credentials and customer support	No material disruption	Continuity report
Instrument	Unify telemetry and margin records	Workload-level reconciliation	Baseline economics
Standardise	Establish canonical workload and result model	Semantic test pass	Architecture approval
Migrate	Move selected customers and adapters	Acceptance and rollback	Migration release
Consolidate	Remove duplicate systems and cost	Realised savings	Value ledger update

Proposed controlled integration plan.

Table 7. Board approval thresholds

Decision area	Green evidence	Amber condition	Red condition
Customer control	Repeated governed workflows and renewal	Useful pilot with conversion plan	Usage without payment or decision use
Portability	Tested core plus provider extensions	Costed adapter gaps	Lowest-common-denominator claims only
Suppliers	Diversified access and transferable terms	Concentrated but replaceable	Critical non-transferable dependency
Economics	Reconciled software gross profit	Margin improvement plan	Pass-through activity valued as software
Security	Controlled multi-provider credentials and lineage	Costed remediation	Unmanaged secrets or customer-data rights
Valuation	Achieved value separated from options	Wide explicit scenarios	Platform label substitutes for evidence

Proposed decision framework.

Frequently asked questions

Q: What should an acquirer value first in quantum cloud middleware? A: Value the controlled customer workflow and its durable gross profit. Test orchestration, portability, telemetry, governance, retention and supplier economics before applying a platform premium.

Q: When does a multi-provider application programming interface create switching cost? A: Switching cost becomes credible when customers rely on governed policy, workflow state, cost controls, result lineage and integrations that remain valuable across providers. A thin routing interface may be easy to replace.

Q: How should pass-through quantum compute be treated? A: Separate compute resale from software and services. Review principal-versus-agent accounting, supplier invoices, credits, commitments, support and cash contribution. Apply valuation to sustainable gross profit and controlled relationships.

Q: Does one interface make workloads portable? A: One interface can reduce development effort. Portability also requires compatible semantics, provider-specific extensions, performance testing, result lineage and a documented migration path.

Q: Which supplier risks matter most? A: Measure spend, workload, feature, customer and geographic concentration. Review pricing, service levels, roadmap access, direct-sales behaviour, data processing, termination and replacement.

Q: What evidence supports a roll-up synergy? A: Identify the customer, product, baseline, owner, cost, timing and measurable outcome. Validate adapter consolidation, cross-selling and margin improvement through contracts, telemetry and realised cash.

Q: Which transaction structures address uncertain platform value? A: Use upfront consideration for controlled assets and achieved economics. Use holdbacks and contingent consideration for customer retention, provider diversification, portability, margin conversion and integration milestones.

Q: What should the board require before approval? A: Require a customer-outcome thesis, complete platform map, reproducible technical review, workload-level unit economics, customer and supplier concentration, contract and security evidence, explicit valuation scenarios and a controlled integration plan.

References

[1] Amazon Web Services, How Amazon Braket works. https://docs.aws.amazon.com/braket/latest/developerguide/braket-how-it-works.html

[2] Amazon Web Services, Amazon Braket supported regions and devices, 2026. https://docs.aws.amazon.com/braket/latest/developerguide/braket-devices.html

[3] Amazon Web Services, Working with Amazon Braket Hybrid Jobs. https://docs.aws.amazon.com/braket/latest/developerguide/braket-jobs.html

[4] Amazon Web Services, Document history for the Amazon Braket Developer Guide, 2026. https://docs.aws.amazon.com/braket/latest/developerguide/doc-history.html

[5] Microsoft, Submit jobs to Azure Quantum with Azure CLI, 2026. https://learn.microsoft.com/en-us/azure/quantum/how-to-submit-jobs-azure-cli

[6] IBM Quantum, Quantum Compute client and runtime service. https://quantum.cloud.ibm.com/docs/en/api/qiskit-ibm-runtime/runtime-service

[7] IBM Quantum, Introduction to Quantum Compute execution modes. https://quantum.cloud.ibm.com/docs/en/guides/execution-modes

[8] IBM Quantum, Introduction to IBM Quantum primitives. https://quantum.cloud.ibm.com/docs/en/guides/qiskit-runtime-primitives

[9] IBM Quantum, Run jobs in a batch. https://quantum.cloud.ibm.com/docs/guides/run-jobs-batch

[10] QIR Alliance, Quantum Intermediate Representation specification. https://www.qir-alliance.org/

[11] OpenQASM, OpenQASM 3 specification. https://openqasm.com/

[12] PennyLane, Devices and quantum hardware plugins. https://docs.pennylane.ai/en/stable/introduction/interfaces.html

[13] NVIDIA, CUDA-Q documentation. https://nvidia.github.io/cuda-quantum/latest/

[14] National Institute of Standards and Technology, SP 800-145 The Definition of Cloud Computing. https://csrc.nist.gov/pubs/sp/800/145/final

[15] National Institute of Standards and Technology, SP 800-207 Zero Trust Architecture. https://csrc.nist.gov/pubs/sp/800/207/final

[16] U.S. Department of Justice and Federal Trade Commission, 2023 Merger Guidelines, Guideline 9. https://www.justice.gov/atr/merger-guidelines/applying-merger-guidelines/guideline-9

[17] U.S. Department of Justice and Federal Trade Commission, 2023 Merger Guidelines overview. https://www.justice.gov/atr/merger-guidelines/overview

[18] IFRS Foundation, IFRS 15 principal-versus-agent considerations. https://www.ifrs.org/news-and-events/updates/ifric/2021/ifric-update-november-2021/

[19] IFRS Foundation, IFRS 3 Business Combinations. https://www.ifrs.org/issued-standards/list-of-standards/ifrs-3-business-combinations/

[20] IFRS Foundation, IAS 38 Intangible Assets. https://www.ifrs.org/issued-standards/list-of-standards/ias-38-intangible-assets/

[21] IFRS Foundation, IFRS 13 Fair Value Measurement. https://www.ifrs.org/issued-standards/list-of-standards/ifrs-13-fair-value-measurement/

[22] International Valuation Standards Council, IVS 210 Intangible Assets. https://www.ivsc.org/standards/

[23] IonQ, Annual Report on Form 10-K for the year ended 31 December 2025. https://www.sec.gov/Archives/edgar/data/1824920/000119312526071562/ionq-20251231.htm

[24] Rigetti Computing, Annual Report on Form 10-K for the year ended 31 December 2025. https://www.sec.gov/Archives/edgar/data/1838359/000110465926023454/rgti-20251231x10k.htm

[25] D-Wave Quantum, Annual Report on Form 10-K for the year ended 31 December 2025. https://www.sec.gov/Archives/edgar/data/1907982/000190798226000026/qbts-20251231.htm

[26] National Institute of Standards and Technology, SP 800-218 Secure Software Development Framework. https://csrc.nist.gov/pubs/sp/800/218/final

[27] National Institute of Standards and Technology, Cybersecurity Supply Chain Risk Management Practices. https://csrc.nist.gov/pubs/sp/800/161/r1/final

[28] European Commission, Data Act and cloud switching. https://digital-strategy.ec.europa.eu/en/policies/data-act

[29] UK Competition and Markets Authority, Cloud services market investigation. https://www.gov.uk/cma-cases/cloud-services-market-investigation

[30] Federal Trade Commission, Hart-Scott-Rodino final rule and merger review. https://www.ftc.gov/legal-library/browse/rules/premerger-notification-reporting-hart-scott-rodino-act

[31] National Institute of Standards and Technology, SP 500-291 Cloud Computing Standards Roadmap. https://www.nist.gov/publications/nist-cloud-computing-standards-roadmap

[32] FinOps Foundation, FinOps Framework. https://www.finops.org/framework/

About the Author

Chennakeshav (CK) is a corporate finance and investment banking executive with 25+ years of global experience in deal origination, structuring and execution across M&A, growth capital and corporate strategy. He has led value-creation mandates for founders, corporates and funds — bridging the boardroom view to hands-on execution and close.

His career spans Morgan Stanley, HSBC, Lloyds Banking Group, EWEC, ADQ portfolio companies and Emirates Growth Fund, across TMT, real estate, fintech, deeptech, cleantech, infrastructure and energy. He has partnered with C-suite leaders, private equity and venture funds, sovereign wealth funds and family offices to finance complex fund raises and scale-up ventures, and has led M&A due diligence, post-merger integration and business-transformation initiatives to create value.

At Matchpoint Partners he is Managing Partner, leading the firm's corporate finance, M&A and capital-raising practice. He holds an MBA from London Business School, an engineering degree from VTU and a Master of Laws (LLM, in progress) from UCL London.

An active start-up mentor, CK mentors at Techstars, DIFC FinTech Hive, Startup Grind, Founder Institute and IN5, serves as Entrepreneur Mentor in Residence (EMiR) at London Business School, and judges the Entrepreneurship World Cup.

https://www.linkedin.com/in/ckadya/

https://www.matchpoint-partners.com/team/ck-adya.html