Watts or Wallets? Pricing Performance and Software Reach in AI Accelerator M&A

A Transaction Framework for Benchmarks, Compiler Maturity, Customer Migration and Platform Value

Authored by Chennakeshav Adya

Independent Researcher

September 2026

Abstract

Artificial-intelligence accelerators are sold through hardware specifications, benchmark results, software claims and customer roadmaps. Transaction value depends on a more demanding question: whether a target can convert silicon, software and system capability into repeatable customer economics. Peak arithmetic throughput can coexist with weak utilisation, limited model coverage, immature compilers, unstable kernels, difficult deployment and expensive migration. A buyer that values headline performance without reconstructing the complete serving stack can pay for theoretical capacity that customers cannot use economically.

This paper develops a transaction framework for acquisitions and investments involving AI accelerator designers, inference-chip companies, compiler businesses, accelerator software platforms and adjacent system assets. It connects workload-specific performance, latency, accuracy, power, memory, networking, software maturity, customer conversion and migration cost to stand-alone value and buyer-specific synergy. It also addresses platform neutrality, interoperability, supply dependencies, roadmap risk, competition review and post-closing controls.

The analysis draws on MLCommons benchmark methodology, public filings and technical disclosures from NVIDIA, AMD and Intel, public cloud and accelerator documentation, open compiler and model-exchange projects, Open Compute Project specifications, semiconductor-policy sources, competition guidance and accounting standards [1-50]. NVIDIA describes a technology stack spanning hardware, networking, systems, software libraries and developer tools [6-10]. AMD describes data-centre accelerators, ROCm software, customer qualification and rapid product cadence [11-14]. MLCommons distinguishes workloads, systems, scenarios, accuracy constraints and availability categories [1-5]. These sources support diligence design. They do not establish the capability, adoption or value of an unidentified target.

An illustrative transaction considers an accelerator-platform company. Management assumes USD 760 million of annual revenue, USD 145 million of EBITDA before normalisation, 64 megawatts of deployed customer capacity, four material customer programmes, USD 280 million of remaining software and system investment, and a stand-alone enterprise value of USD 1.68 billion. Management also assumes potential buyer value from procurement, distribution, workload optimisation and faster customer migration. Every amount, probability and valuation result is a management assumption used solely to demonstrate the method.

The central conclusion is that an accelerator premium should be paid for reproducible workload economics and durable software reach. Performance per watt, performance per dollar and rack density matter when the measurements reproduce under customer workloads, accuracy thresholds, realistic concurrency and complete system costs. Software value matters when compilers, libraries, frameworks and operational tools reduce deployment effort and support recurring use. The valuation should deduct *migration cost, roadmap capital, customer concentration and platform risk before recognising buyer-specific synergy.*

JEL Classification: G24, G32, G34, L22, L63, O31, O32

Keywords: AI accelerators, semiconductor M&A, compiler maturity, software ecosystem, customer migration, performance per watt, performance per dollar, valuation

Introduction

AI accelerators compete across silicon architecture, memory, networking, packaging, systems, compilers, libraries, frameworks, orchestration and developer support. Public filings describe competition based on performance, energy efficiency, integration, ease of use, workload optimisation, software ecosystems, roadmaps and supply [6-16]. MLCommons benchmark rules show why comparison requires defined models, scenarios, accuracy constraints, system descriptions and availability status [1-5]. A single peak specification cannot capture this operating system.

The transaction problem is therefore economic and technical. A buyer needs to know which workloads the target can run, at what latency and throughput, with what accuracy, power, memory and engineering burden, and whether those results can be reproduced by customers. It must also know whether the software stack supports new models, whether customers can migrate from incumbent platforms, whether deployments can scale across clusters and whether the business can deliver its roadmap through supply and capital constraints.

This paper is designed for strategic buyers, private-equity investors, corporate-development teams, founders, lenders and investment committees. It provides a system for diligence, valuation, deal terms and integration. It does not provide engineering, legal, tax, accounting, regulatory or investment advice. Each transaction requires current specialist review of products, contracts, benchmarks, source code, intellectual property, supply arrangements, security, export controls and customer evidence.

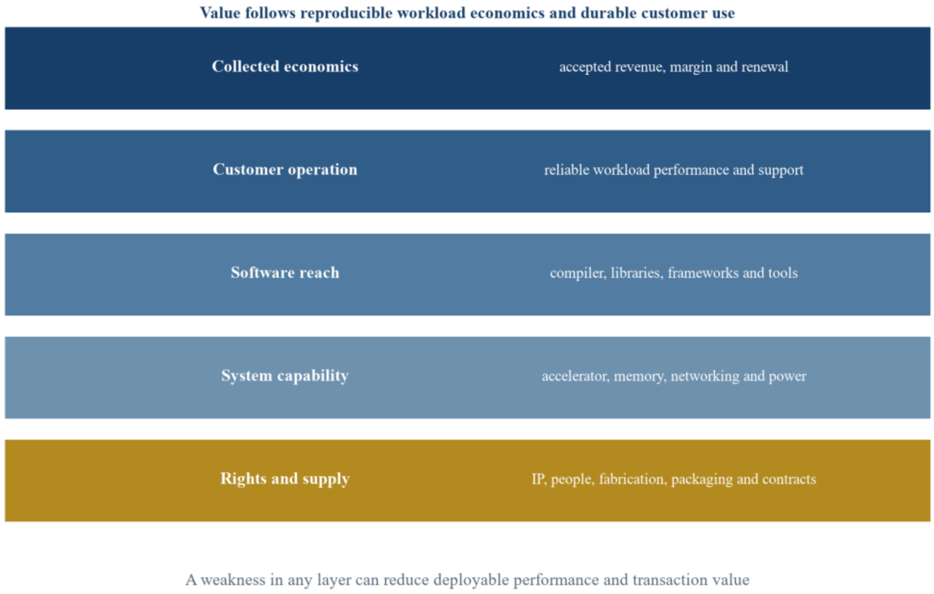
1 Define the acquisition perimeter before pricing the platform

An accelerator target may own processor architecture, chiplets, memory interfaces, boards, systems, compiler technology, kernels, runtime software, model-serving tools or vertical applications. The legal entity may control only part of the product customers use. The buyer should map the full system while identifying the assets and obligations that will transfer at closing.

The asset map should connect patents, design files, verification environments, firmware, compilers, libraries, model integrations, benchmark scripts, telemetry, customer contracts, cloud images, supply agreements and trained teams. Each asset should have an owner, revision, dependency and evidence of use. A performance demonstration can show potential. It should remain distinct from a reproducible customer deployment with contracted economics.

The acquisition perimeter should also distinguish products from services. Engineering support, model conversion, kernel optimisation and deployment assistance can accelerate adoption while hiding software gaps. Revenue that depends on bespoke engineering has different durability and margin from revenue generated through a stable platform. The model should identify the labour, tooling and customer-specific work required for every material deployment.

Figure 1 The AI accelerator transaction value chain



Proposed diligence architecture; value depends on conversion from controlled technology to accepted customer economics.

Table 1 Accelerator asset and evidence map

Value layer	Claimed asset	Minimum evidence	Principal transaction risk
architecture	processor, memory and interconnect design	controlled design, verification history, measured silicon and ownership map	benchmark advantage depends on an unreleased configuration
software	compiler, runtime, kernels and libraries	source control, release history, test coverage and workload support	customer performance depends on manual optimisation
systems	boards, racks, networking, cooling and orchestration	qualified bill of materials, telemetry and support records	component bottlenecks erase chip-level advantage
adoption	design wins, cloud availability and production use	contracts, usage, acceptance and renewal evidence	evaluations are counted as recurring demand
economics	price, service burden, margin and cash	invoice, cost, credits, support hours and collection	hardware revenue conceals uneconomic enablement work

Proposed diligence structure; technical claims should reconcile to controlled repositories and customer records.

2 Benchmark customer workloads rather than peak specifications

Peak operations per second describe a theoretical rate under defined precision and conditions. Customer value depends on the workload, model, batch size, sequence length, accuracy target, latency requirement, concurrency, memory footprint and software configuration. A benchmark should therefore be treated as a controlled experiment, not a universal statement about product value.

MLCommons separates system types, availability categories, benchmark models and deployment scenarios [1-5]. Its documentation describes accuracy validation, latency tracking, load generation and result review. These controls provide a useful diligence pattern. The buyer should preserve the complete system description, benchmark code, compiler flags, model version, data, power boundary, run logs and result checks for every claimed advantage.

The test plan should include representative customer workloads. Training, batch inference, interactive inference, retrieval, recommendation, vision and scientific computing can stress different resources. Performance may change with model size, sparsity, precision, memory bandwidth, communication pattern and requested service level. A target that leads one workload can remain commercially valuable, but the valuation should follow the relevant addressable workload rather than extrapolate across the whole AI market.

Benchmark reproducibility is a governance question. The buyer should rerun material claims on controlled systems, compare vendor-optimised and portable implementations, and test sensitivity to software versions. It should document warm-up, caching, quantisation, batching and excluded failures. Results should be presented as a range with the conditions required to reproduce them.

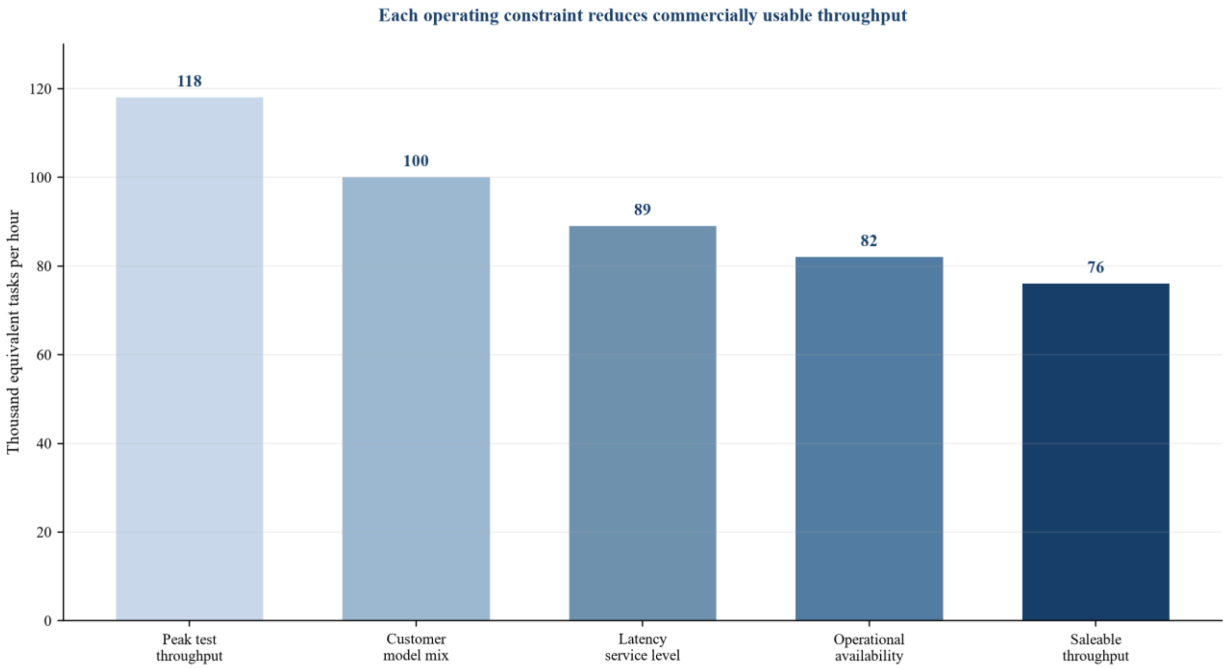
Benchmark provenance should extend to the underlying models and data. The diligence team should record whether weights are public, licensed or customer-controlled; whether the model was modified; which accuracy or quality tests were applied; and whether preprocessing or postprocessing work is included in the measured interval. It should retain container images, dependency manifests, firmware, drivers, compiler versions and machine configuration. A result that cannot be recreated after a routine software update is weak transaction evidence. The buyer should also test a neutral implementation where practical. Vendor-tuned code can show the platform's attainable performance, while a portable implementation can reveal the engineering burden required to reach it. Both views matter because customer economics depend on the path from ordinary deployment to optimised production.

3 Reconstruct performance under load and over time

AI serving systems operate across a curve. High concurrency may improve aggregate throughput while increasing per-user latency. Low latency may require underutilised capacity. Long prompts, large models, retrieval steps and variable outputs can change the balance. MLCommons Endpoints describes measurement across throughput, interactivity, time to first token and concurrency [3]. A transaction model should connect this curve to customer service levels and realised price.

Management assumes an illustrative system delivers 118 thousand equivalent tasks per hour at a high-throughput operating point. It assumes a 15% reduction for customer model mix, a further 11% reduction for latency commitments, 8% for operational availability and 6% for software and data overhead. The resulting saleable throughput is approximately 76 thousand equivalent tasks per hour. These assumptions demonstrate the method and do not describe an identified company.

Figure 2 Illustrative conversion of peak accelerator throughput to saleable customer throughput



Volumes and conversion rates are management assumptions used solely to demonstrate the method.

Table 2 Benchmark evidence gates

Evidence state	Required evidence	Valuation treatment	Common overstatement
specification	published architecture and supported precision	technical context only	peak rate treated as application throughput
laboratory run	controlled system, code, logs and accuracy	evidence for the tested configuration	selected run treated as normal production
reproducible benchmark	independent rerun across relevant scenarios	workload-specific performance range	one model applied across the addressable market
customer pilot	target workload, service level and operating record	programme option after remaining work	proof of concept counted as recurring use
accepted production	contracted use, telemetry, invoice and collection	evidenced operating value	booked capacity treated as consumed demand

Proposed classification; the buyer should use target-specific workloads, systems and service levels.

The buyer should test degradation as the system ages and software evolves. New model architectures can expose unsupported operators, memory limitations or communication overhead. Driver and framework updates can improve performance while introducing regression risk. Performance evidence should therefore be dated, versioned and connected to the product roadmap.

Operational availability should include maintenance, failures, deployment errors and recovery. A system with high benchmark speed can deliver weak economics when utilisation is interrupted or support demands are high. The model should reconcile scheduled capacity, successful jobs, billable units, service credits and collections.

4 Price performance per dollar, per watt and per rack

Customers buy useful work within constraints. Performance per dollar captures acquisition or rental price. Performance per watt captures energy cost and power availability. Performance per rack captures facility density, networking and cooling. Each measure can change the preferred platform. The buyer should identify which constraint governs each customer segment and location.

The cost boundary should include accelerators, CPUs, memory, networking, storage, chassis, power conversion, cooling, software, support and migration. A lower-priced chip can create a higher system cost when it requires more nodes, engineering or networking. A high-priced platform can produce strong customer economics when utilisation, software coverage and time to deployment are superior. The valuation should be based on total delivered economics rather than component list price.

Table 3 Illustrative accelerator unit-economics bridge

Item	Central case	Downside case	Diligence focus
realised platform revenue	USD 46,000	USD 39,000	contract duration, usage and price resets
silicon, memory and board cost	USD 20,500	USD 22,800	yield, allocation, mix and warranty
networking, software and support	USD 9,200	USD 12,700	system content, licences and engineering burden
migration and deployment allowance	USD 3,300	USD 5,800	model conversion, tuning and customer acceptance
contribution before fixed cost	USD 13,000	negative USD 2,300	sustainable utilisation and service intensity

All values are management assumptions per annual equivalent deployed accelerator and demonstrate the method only.

Power should be measured at the relevant boundary. Chip power, board power, server power and facility power are different measures. Cooling, networking and idle capacity can materially affect cost. The buyer should examine metered evidence and the operating point at which it was recorded. It should also test whether performance improvements rely on aggressive power settings that customers cannot sustain.

Facility constraints can create strategic value. An accelerator that delivers more accepted work inside an existing power envelope can defer data-centre capital. That value depends on workload reproducibility, system availability and integration. It belongs partly to the customer or buyer and should not automatically be capitalised into the target's stand-alone value.

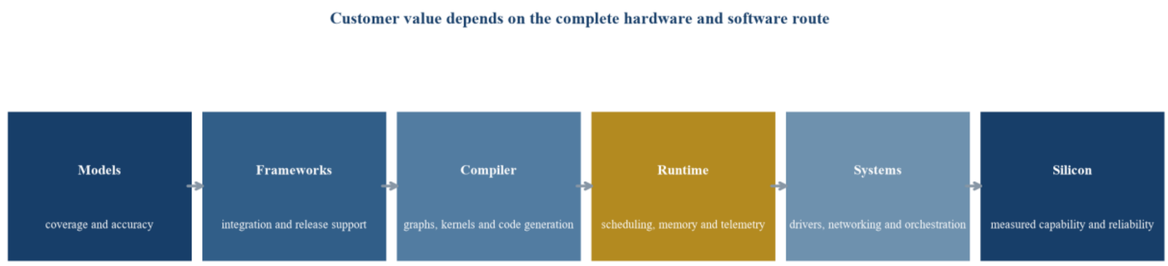
5 Value compiler maturity and software reach as operating assets

NVIDIA describes CUDA and a broad set of libraries, frameworks, SDKs and APIs as part of its technology stack [6-10]. AMD describes ROCm, customer support and an expanding data-centre portfolio [11-14]. OpenXLA, LLVM, ONNX and major frameworks illustrate the importance of compiler infrastructure, model representation and portability [21-25]. These sources show the dimensions of software value. They do not prove equivalent maturity across platforms.

Compiler diligence should cover front ends, intermediate representations, graph optimisation, code generation, kernel selection, debugging, profiling, numerical correctness and hardware scheduling. The buyer should review supported operators, model coverage, release cadence, regression history and time required to enable new architectures. It should identify which performance comes from reusable compiler capability and which comes from hand-tuned customer work.

Software reach includes documentation, installation, cloud images, containers, orchestration, security updates, observability and support. Developer adoption should be evidenced through active production use, releases, issue resolution, contributor quality and customer retention. Download counts, repositories and community activity can provide context. They should not be treated as recurring economics without customer evidence.

Figure 3 The accelerator software value stack



Proposed evidence chain; each layer should be tested for portability, reproducibility and customer use.

Software maturity should enter valuation through measurable economic effects. It can improve time to deployment, utilisation, customer conversion, support cost and renewal. The model should avoid a broad platform premium unsupported by programme evidence. It should link each software capability to a customer cohort, remaining investment and observable outcome.

Source-code diligence should examine architecture, maintainability, testing, security, release automation and third-party obligations. The buyer should identify code that is owned, permissively licensed, restrictively licensed, customer-provided or dependent on confidential interfaces. Build reproducibility should be tested from a clean environment. Critical components should have named maintainers, review history and recovery plans. Technical debt can be economically material when each new model or chip requires extensive manual intervention. The valuation should distinguish remediation required to sustain current revenue from discretionary investment intended to expand the platform. Open-source participation can strengthen adoption and recruitment, while creating compliance, disclosure and governance obligations that need separate review.

6 Measure migration cost before assuming customer conversion

Migration involves more than recompiling code. Customers may need to convert models, replace unsupported operations, retune kernels, change precision, validate accuracy, rebuild containers, modify schedulers, retrain teams and redesign monitoring. They may also need to duplicate infrastructure during transition. The buyer should measure this burden by workload and customer.

The migration plan should start with an inventory of models, frameworks, custom operators, data pipelines, serving systems, security controls and service levels. Each item should receive a portability assessment, remediation plan, responsible owner, test and acceptance gate. The cost model should include customer engineering, target support, delayed use and parallel operation.

Switching cost can support retention while limiting growth. A target may have strong customers because it solves a valuable problem or because leaving is expensive. The buyer should distinguish product advantage from dependency. It should also assess whether acquisition by a platform owner changes customer trust, neutrality or willingness to share workloads.

Migration incentives should be treated as acquisition cost when they are required to win business. Free engineering, credits, hardware discounts and capacity guarantees can accelerate adoption while reducing realised price. The revenue bridge should show gross bookings, incentives, accepted use and collected cash.

Migration capacity should be modelled as a constrained delivery system. Specialist engineers, access to customer models, validation environments and production windows may limit the number of programmes that can move at the same time. A large pipeline can therefore convert slowly even when customers express interest. The buyer should estimate engineering hours, elapsed time, acceptance cycles and support intensity by workload type. It should identify reusable tools and learning that reduce later migration cost. The forecast should cap conversion at the demonstrated delivery capacity until funded hiring, automation or partner support is available. This approach prevents a sales pipeline from being converted into revenue faster than the target can technically deliver and customers can operationally accept.

7 Test design wins, customer concentration and usage quality

An accelerator design win can mean technical evaluation, approved-vendor status, reserved capacity, initial deployment or committed volume. The diligence vocabulary should assign a defined evidence requirement to each state. A press release or memorandum of understanding can support strategic context. It should not be valued as contracted cash.

The buyer should reconcile pipeline records with contracts, orders, telemetry, invoices, credits and collections. It should identify the customer entity, workload, product version, deployment site, service level, price, minimum commitment, cancellation rights and acceptance conditions. Forecast volume should be separated from installed, available, consumed and billable capacity.

Concentration should be measured across customers, workloads, cloud channels, system partners and common economic parents. Several programmes can depend on one hyperscaler or one model family. A customer can also have leverage through qualification, capacity funding or software contribution. The valuation should test the loss, delay or repricing of each material relationship.

Table 4 Customer adoption and valuation treatment

Programme state	Evidence	Valuation treatment	Required protection
evaluation	test plan, system access and responsible teams	limited option value	cost cap and decision date
technical selection	written selection and remaining conditions	programme option after migration cost	objective acceptance gates
deployment	installed system, telemetry and support plan	probability-weighted operating value	capacity and service obligations
accepted use	service-level compliance, invoice and collection	evidenced cash-flow value	renewal and pricing controls
scaled renewal	recurring use across periods or workloads	durable customer value after concentration test	continuity and change-of-control plan

Proposed framework; probabilities and conversion periods require target-specific evidence.

Usage quality matters. A deployment may run below planned utilisation because models, data or customer demand are delayed. It may run at high utilisation with weak margins because support and power costs are underpriced. The investment committee should receive customer-level unit economics rather than a single installed-capacity figure.

8 Map fabrication, memory, packaging and networking dependencies

Accelerator performance depends on fabrication, packaging, HBM, substrates, networking, system integration and power delivery. Open Compute Project specifications illustrate modular accelerator and system interfaces [26]. UCIe, memory and interconnect standards provide additional context [27-29].

Public foundry and packaging disclosures describe continuing investment and technical complexity [30-32]. The target's roadmap should be tested against these dependencies.

The buyer should inspect wafer agreements, packaging allocation, memory supply, board manufacturing, component qualification, change control, priority, pricing, minimum purchase, warranty and second-source status. A supplier name on a plan does not establish accessible capacity. The model should use executed rights, dated readiness and demonstrated yield.

Networking and communication can become the system bottleneck as clusters scale. The test plan should measure collective operations, congestion, failure recovery and workload scaling. A chip-level advantage can disappear when software or fabric overhead rises. The valuation should therefore include system-level performance and the capital required for the promised configuration.

Supply commitments can create both value and liability. Prepayments and take-or-pay obligations may secure capacity while consuming liquidity if demand moves. Customer-funded capacity can reduce capital needs while creating priority, refund or exclusivity terms. The buyer should show each commitment by product, period, counterparty and change-of-control consequence.

9 Value the roadmap as a sequence of evidence gates

Accelerator roadmaps move through architecture, simulation, tape-out, first silicon, bring-up, software enablement, system qualification and customer acceptance. Each gate has different probability, remaining cost and timing. A roadmap slide should not receive full operating value before evidence supports conversion.

The buyer should maintain one integrated hardware and software roadmap. Silicon delivery without compiler, framework and system readiness can delay revenue. Software may mature after launch and improve performance. The forecast should state the version required for each customer programme and the resources needed to deliver it.

Roadmap value can be modelled as a set of programme options. Each option should have a defined product, workload, customer cohort, expected cash, probability, remaining investment and earliest decision date. Shared costs and dependencies should be allocated once. The model should prevent the same future capability from appearing in both terminal growth and separate option value.

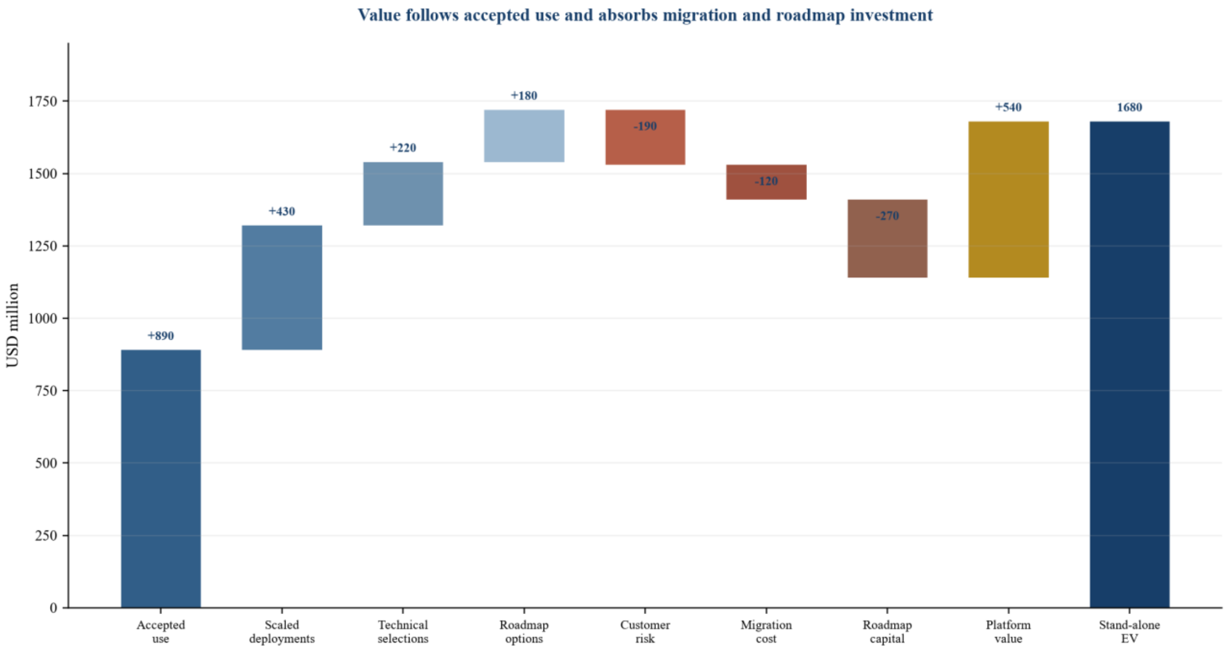
Terminal value should reflect the capability to deliver successive products, not one favourable generation. The buyer should assess architecture reuse, team continuity, verification assets, software portability, supplier access and customer trust. A rapid cadence can support value when prior gates have been achieved with controlled cost and quality.

10 Build the valuation from accepted workload economics

The stand-alone valuation should begin with customer-level cash flows. Revenue should be driven by accepted deployments, consumed or contracted units, realised price and renewal. Costs should include silicon, memory, systems, software, support, migration, warranty, capital and working capital. Scenario analysis should combine performance, utilisation, price, power and roadmap timing.

Management assumes five accepted customer cohorts with a present value of USD 890 million, four scaled deployment options worth USD 430 million, four technical selections worth USD 220 million and four roadmap options worth USD 180 million. It then deducts USD 190 million for customer concentration, USD 120 million for migration support and USD 270 million of remaining roadmap capital. A platform allowance of USD 540 million produces the assumed stand-alone enterprise value of USD 1.68 billion. All amounts are management assumptions.

Figure 4 Illustrative enterprise-value bridge for an AI accelerator platform



All amounts are management assumptions in USD millions and do not describe an identified company.

Table 5 Illustrative programme valuation schedule

Programme state	Cohorts	Illustrative probability	Average present value per cohort	Probability-weighted value
accepted use	5	100%	USD 178m	USD 890m
scaled deployment	4	80%	USD 134m	USD 430m
technical selection	4	50%	USD 110m	USD 220m
roadmap option	4	25%	USD 180m	USD 180m
gross programme value	17	mixed	mixed	USD 1,720m

Values and probabilities are management assumptions for method demonstration only.

Market multiples can provide a reasonableness check after comparability is established. Revenue quality, hardware content, software maturity, gross margin, capital intensity, customer concentration and roadmap risk differ materially across accelerator businesses. A platform multiple should not be applied merely because the target uses AI terminology.

The model should separate stand-alone value from buyer-specific value. Distribution, supply access, cloud reach, system integration and software optimisation may create synergy. Their value depends on buyer resources, implementation cost, customer response and competition constraints. The seller should not receive full payment for synergy that only the buyer can produce.

Financing analysis should reflect the volatility and capital needs of the platform. Debt capacity should be based on recurring, accepted cash flow after supply commitments, warranty, support and roadmap investment. Backlog, reservations and customer forecasts should be classified by enforceability and cancellation rights. A lender may require liquidity for tape-outs, inventory, capacity deposits and customer migration before revenue is collected. The downside case should combine delayed deployment, weaker realised price, higher support cost and continued roadmap spending because these pressures can occur together. Acquisition finance should include covenant headroom and funding for the integration plan rather than assuming that headline EBITDA is immediately distributable.

11 Build buyer-specific synergy with accountable delivery plans

Synergy can arise from procurement, packaging allocation, networking, cloud distribution, software integration, customer access and reduced duplicated cost. Each initiative should have a baseline, owner, resources, timing, dependency and measurable cash outcome. A label such as ecosystem leverage is insufficient for valuation.

Technical synergy should be tested through defined workloads. Combining a target accelerator with a buyer compiler, interconnect or system can improve performance, utilisation or deployment time. The model should include integration cost and the risk that customers reject a closed or changed platform. Results should be measured against the signing baseline.

Revenue synergy requires customer evidence. A buyer channel may expand access while creating conflicts with customers that compete with the buyer. The model should identify customers, offers, sales cycles, migration support and capacity. Broad percentage uplifts should remain outside the central case unless supported by programme plans.

Cost synergy should preserve critical capability. Removing duplicated engineering without understanding compiler, verification, support and customer dependencies can damage the platform. The integration plan should distinguish removable corporate cost from product work required for roadmap and reliability.

12 Choose a transaction structure that matches evidence maturity

A full acquisition can support integrated product and distribution decisions while concentrating roadmap and customer risk at closing. A staged acquisition can link ownership and consideration to silicon, software or customer gates. A minority investment can secure learning and commercial rights while preserving neutrality. A joint venture can combine complementary assets while creating governance complexity. A licensing or supply agreement can provide access without acquiring the company.

Table 6 Transaction structure against accelerator evidence maturity

Structure	Suitable evidence state	Buyer control	Principal protection	Principal limitation
full acquisition	accepted use, durable rights and integration plan	high	conditions, indemnity, retention and covenants	upfront exposure to roadmap and concentration
staged acquisition	strong technology with remaining customer or software gates	high after milestones	milestone consideration and call or put mechanics	future pricing and governance complexity
minority investment	developing platform with learning value	limited	information, consent and participation rights	limited authority over capital and execution
joint venture	complementary silicon, software or distribution assets	shared	contribution, field, funding and deadlock rules	divided authority and IP leakage risk
licence or supply agreement	need for access with limited ownership rationale	contractual	scope, service levels, continuity and audit	supplier remains responsible for execution

Proposed decision framework; legal, tax, accounting and regulatory consequences require specialist advice.

Contingent consideration can bridge uncertainty when the metric is measurable and buyer control is addressed. Relevant gates may include benchmark reproducibility, compiler coverage, accepted deployment, usage, gross margin or roadmap delivery. The agreement should define configuration, evidence, measurement period, customer changes, capital commitments, accounting, audit and dispute resolution.

Governance should protect the asset before full control. Information rights, funding obligations, roadmap decisions, IP use, customer neutrality and related-party transactions should match the structure. The buyer should avoid receiving competitively sensitive customer information beyond its legitimate rights.

13 Address competition, interoperability and platform neutrality

The US Merger Guidelines discuss transactions involving platforms, complements, interoperability and access to inputs [37-39]. An accelerator acquisition can affect developers, cloud providers, customers, system partners and competing hardware. The buyer should identify where the target supports multi-platform use and whether the transaction could change incentives to maintain that support.

Competition diligence should examine market definition, customer alternatives, switching cost, roadmap timing, supply access, developer tools and data. A small current revenue base does not remove forward-looking issues when the target could reduce dependence on an established platform. The analysis should be led by qualified counsel using current law and facts.

Remedies or commitments can affect value. Requirements concerning interoperability, licensing, information separation, supply, customer neutrality or non-discrimination can limit planned integration or synergy. The valuation should include the cost, delay and strategic effect of credible outcomes rather than treating approval as a binary event.

Platform neutrality can itself be an asset. Customers may value a compiler, orchestration layer or model tool because it supports several accelerators. Acquisition by one hardware supplier can weaken that proposition. The buyer should test retention with customers and partners and preserve governance where neutrality supports cash flow.

Data, security and export controls can shape the integration perimeter. Customer workloads may contain confidential models, training methods, personal data or regulated information. Benchmark artefacts and telemetry may be contractually restricted. Source repositories can contain controlled technical information, cryptographic material or third-party code. The buyer should map where these assets are stored, who can access them, how they cross borders and which approvals apply. Integration plans should preserve least-privilege access, logging, segregation and incident response. Any planned transfer of teams, tools or technology should be reviewed against current export and sanctions rules. Costs and delays arising from required controls belong in the transaction model and closing plan.

14 Reconcile capital expenditure, working capital and support obligations

Accelerator development requires design, verification, tape-out, software, systems, inventory, qualification and support. The buyer should separate maintenance, committed roadmap, discretionary growth and customer-funded investment. Announced capital should be reconciled to purchase orders, contracts, milestones, payments and usable output.

Inventory risk can rise during generation transitions. Chips, boards and systems may become less valuable when software, customer schedules or new products move. The diligence team should classify inventory by version, customer, qualification, recoverability and cancellation rights. Purchase commitments and supplier deposits should be tested under central and downside demand.

Working capital should include receivables, credits, warranties, support commitments, prepayments and capacity reservations. Hardware shipment does not always equal final acceptance. The cash-flow model should reflect installation, acceptance, usage, service credits and collection.

Capitalised software and development costs require accounting review. The buyer should reconcile economic life with product cadence and customer use. Purchase accounting should identify acquired technology, customer relationships, contracts and goodwill using current standards [43-47]. Accounting classifications do not replace the operating valuation.

15 Protect customer trust and engineering continuity through integration

Integration should preserve roadmaps, release quality, customer confidentiality, security, supply and accountable decision rights. Customers may worry that a strategic buyer will favour its own products, change pricing, reduce portability or access sensitive workloads. Communication should address continuity, support, interoperability and governance using commitments the combined company can deliver.

Key-person risk extends beyond executives. Architecture, verification, compiler, kernel, framework, support and customer knowledge may sit with small teams. Retention should be connected to role, authority, roadmap and knowledge transfer. Documentation should include design decisions, source history, build systems, benchmark artefacts, test coverage and known defects.

Table 7 First-hundred-day accelerator integration controls

Workstream	Required evidence	Day-one control	First-hundred-day outcome
customers	contracts, workloads, service levels and communication map	named relationship owner and confidentiality protocol	verified programme baseline and retention plan
engineering	architecture, source, releases, defects and roadmap	protected teams, repositories and release authority	integrated roadmap with funded gates
benchmarks	configurations, code, logs, accuracy and power boundaries	frozen definitions and independent rerun plan	reproducible customer-relevant performance dashboard
supply	wafer, packaging, memory, systems and purchase commitments	accountable owner for each critical dependency	renewed rights and tested contingency plan
value	signing model, migration cost, synergy and integration funding	one controlled baseline and change log	measured accepted use, cost and net synergy report

Proposed control plan; timing should be adapted to transaction approvals and customer obligations.

Integration sequencing should follow customer and release risk. Immediate changes to compiler, drivers, repositories, build systems or telemetry can create regressions. The buyer should establish which systems can change at day one, which require controlled testing and which should remain separate until a product or customer gate.

16 Use a signing-to-accepted-deployment control model

The interval between signing and accepted customer use can contain substantial value movement. Product schedules, software releases, benchmark results, supply allocation and customer programmes continue to change. The control model should cover the final diligence cut through closing, integration and at least the first audited cycle of accepted deployments and collections.

The model should begin with a frozen signing baseline. It should state each programme, workload, configuration, benchmark, service level, price, supply route, required staff, capital and forecast cash.

Changes should be recorded against the baseline. A software regression, delayed tape-out, lower supply, customer redesign or price reset can affect value before it appears in reported revenue.

Interim covenants should protect material assets and programmes while preserving lawful operating responsibility. The seller should maintain teams, source repositories, release controls, supply rights, customer relationships and ordinary-course capital. Material IP grants, exclusivity, roadmap changes, cancellations or unplanned commitments may require consent, subject to applicable law and negotiated thresholds.

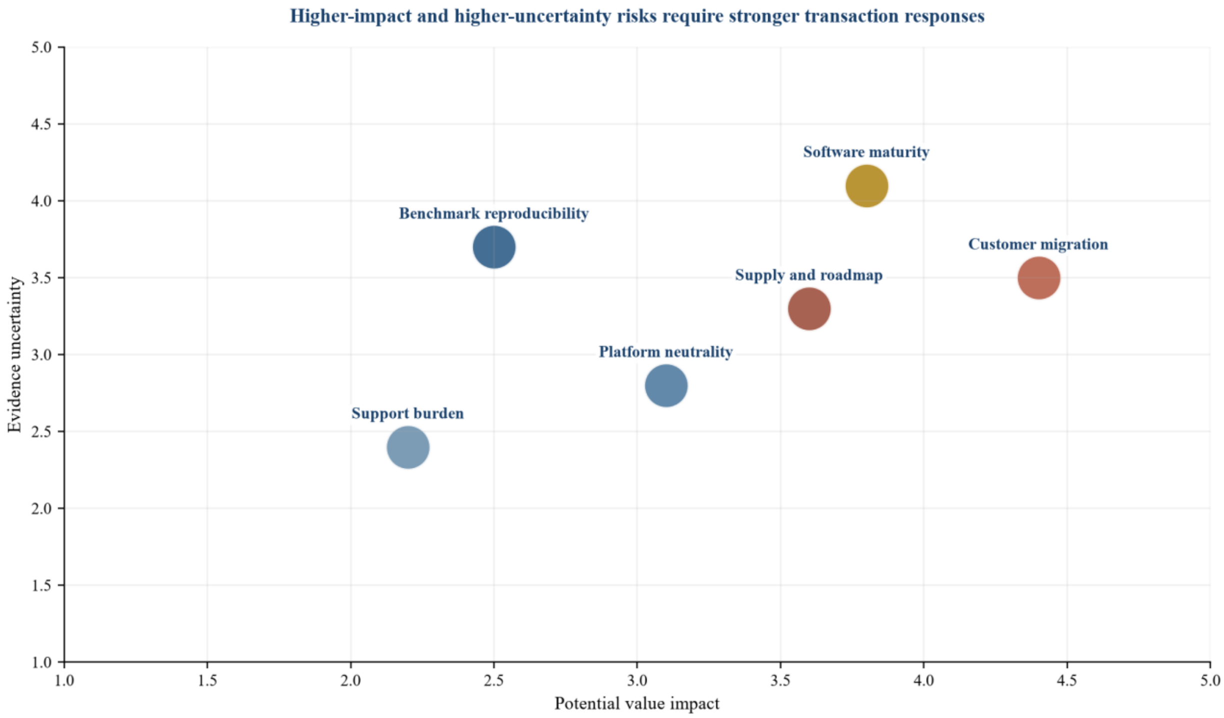
Closing readiness should include tested access to repositories, build systems, signing keys, benchmark artefacts, support systems, supplier contacts and customer escalation. The buyer should know which credentials and data can transfer, which require consent and which must remain segregated. Every unresolved dependency should have an owner and dated action.

17 Translate diligence findings into price, terms and integration actions

Diligence creates value when findings change a decision. Every material finding should be classified as a price adjustment, structural item, closing condition, contractual protection, integration action or monitored risk. The classification should identify evidence, financial exposure, timing, owner and decision.

A price adjustment can address weaker accepted use, unsupported performance, migration cost or required capital. A closing condition can address a customer approval, supply consent or financing. A representation, indemnity or escrow can address an identified exposure. A covenant can preserve teams, repositories, release discipline or supply. An integration action can fix a controllable weakness after closing.

Figure 5 Accelerator transaction risk and response map



Proposed decision map; position and treatment should be calibrated to target-specific evidence.

Representations and warranties should match the nature and duration of the risk. They can cover IP ownership, open-source compliance, benchmark statements, customer contracts, supply rights, security, export controls and financial statements. Insurance can support some contractual risks. It does not replace technical testing, customer evidence or a funded integration plan.

The final investment paper should reconcile headline price, net debt, working capital, contingent consideration, retention, transaction cost and integration funding. It should present stand-alone value, buyer-specific value and consideration on the same basis. It should show how each major diligence finding changed the model or terms.

Conclusion

AI accelerator transactions combine semiconductor, software, system, customer and platform economics. A credible valuation follows a reproducible chain from controlled rights and supply through measured silicon, mature software, customer deployment, accepted use and collected cash. Peak specifications and selected benchmarks provide context. They require workload, accuracy, latency, power and system evidence before they support operating value.

The strongest platform delivers useful work economically across relevant customer conditions. It combines reproducible performance, compiler and library maturity, manageable migration, reliable supply, customer trust and a funded roadmap. Its premium derives from repeatable adoption and durable control rather than one test result or product cycle.

The transaction discipline is practical. Define the asset. Reconstruct performance under load. Price complete-system economics. Test software reach and migration. Verify customer use. Map supply and roadmap dependencies. Value programmes from accepted cash. Separate buyer synergy. Choose terms that match evidence maturity. Maintain one controlled baseline through closing and accepted deployment.

Frequently asked questions

Which performance measure matters most in accelerator valuation?

The relevant measure is customer-accepted workload economics under defined accuracy, latency, concurrency, power and system conditions. Peak specifications, isolated kernels and selected benchmark results remain useful inputs. They require reproducibility and a clear link to addressable customer workloads before they support revenue value.

How should a buyer compare performance per watt and performance per dollar?

The buyer should use the same workload, service level, accuracy and system boundary. Performance per watt should identify whether power is measured at chip, board, server or facility level. Performance per dollar should include hardware, software, networking, support, migration and required capital.

How does compiler maturity enter valuation?

Compiler maturity affects model coverage, deployment time, utilisation, correctness, support cost and customer conversion. Value should be linked to reproducible releases, supported workloads, regression control, customer use and the remaining investment required for the roadmap.

When is software reach a durable moat?

Software reach is durable when customers repeatedly deploy workloads through maintained compilers, libraries, frameworks and operational tools, and when the target can support new models without disproportionate bespoke engineering. Repository activity alone does not establish durable economics.

How should customer migration cost be treated?

Migration cost should be estimated by workload and customer. It includes model conversion, custom operators, tuning, validation, retraining, parallel capacity and incentives. Costs required to win or retain customers reduce programme value unless funded by a separate buyer-specific synergy case.

When is contingent consideration useful?

Contingent consideration can bridge uncertainty over benchmark reproducibility, software coverage, accepted deployment, utilisation, margin or roadmap delivery. The metric should be measurable, the evidence controlled and buyer obligations defined.

What can destroy accelerator value after closing?

Software regressions, roadmap delays, supply loss, customer migration failures, loss of platform neutrality, key-team departure, security incidents and underfunded support can destroy value. The integration plan should assign owners, funding, gates and measurable outcomes before signing.

What should the board monitor after acquisition?

The board should monitor reproducible customer workloads, accepted deployments, utilisation, realised price, power and system cost, software coverage, migration progress, concentration, supply, roadmap gates, integration cost and net synergy against the frozen signing baseline.

References

[1] MLCommons, MLPerf Inference Benchmark Suite, https://docs.mlcommons.org/inference/index_gh/

[2] MLCommons, MLPerf Inference Submission Guide, https://docs.mlcommons.org/inference/submission/

[3] MLCommons, MLPerf Endpoints benchmark, https://mlcommons.org/benchmarks/endpoints/

[4] MLCommons, MLPerf Inference power measurement, https://mlcommons.org/benchmarks/inference-power/

[5] MLCommons, MLPerf Inference v5 language-model benchmarks, https://mlcommons.org/2025/04/llm-inference-v5/

[6] NVIDIA, 2026 Annual Report on Form 10-K, https://www.sec.gov/Archives/edgar/data/1045810/000104581026000021/nvda-20260125.htm

[7] NVIDIA, 2026 annual report and proxy materials, https://www.sec.gov/Archives/edgar/data/1045810/000104581026000038/a2026-annualxreportxwebxfi.pdf

[8] NVIDIA, fiscal 2026 fourth-quarter results, https://www.sec.gov/Archives/edgar/data/1045810/000104581026000019/q4fy26pr.htm

[9] NVIDIA, CUDA platform documentation, https://docs.nvidia.com/cuda/

[10] NVIDIA, TensorRT documentation, https://docs.nvidia.com/deeplearning/tensorrt/

[11] Advanced Micro Devices, 2025 Annual Report on Form 10-K, https://www.sec.gov/Archives/edgar/data/2488/000000248826000018/amd-20251227.htm

[12] Advanced Micro Devices, ROCm documentation, https://rocm.docs.amd.com/

[13] Advanced Micro Devices, data centre and Instinct product disclosures, https://www.amd.com/en/products/accelerators/instinct.html

[14] Advanced Micro Devices, annual reports and filings, https://ir.amd.com/financial-information/annual-reports-and-proxies

[15] Intel, 2024 Annual Report on Form 10-K, https://www.intc.com/filings-reports/annual-report

[16] Intel, Gaudi accelerator documentation, https://docs.habana.ai/

[17] Google Cloud, TPU documentation, https://cloud.google.com/tpu/docs

[18] Amazon Web Services, Trainium documentation, https://docs.aws.amazon.com/trainium/

[19] Microsoft, Maia AI accelerator overview, https://news.microsoft.com/source/features/ai/in-house-chips-silicon-to-service-to-meet-ai-demand/

[20] Meta Engineering, MTIA accelerator programme, https://engineering.fb.com/2023/05/18/data-center-engineering/mtia-v1-meta-training-inference-accelerator/

[21] OpenXLA, compiler project documentation, https://openxla.org/

[22] LLVM Project, compiler infrastructure documentation, https://llvm.org/docs/

[23] ONNX, open model exchange documentation, https://onnx.ai/

[24] PyTorch, compiler documentation, https://pytorch.org/docs/stable/torch.compiler.html

[25] TensorFlow, XLA documentation, https://www.tensorflow.org/xla

[26] Open Compute Project, OCP Accelerator Module Base Specification, https://www.opencompute.org/documents/oai-oam-base-specification-r2-0-v1-0-20230919-pdf

[27] UCIe Consortium, specification resources, https://www.uciexpress.org/specification

[28] PCI-SIG, Compute Express Link resources, https://www.computeexpresslink.org/

[29] JEDEC, high-bandwidth memory standards resources, https://www.jedec.org/

[30] Taiwan Semiconductor Manufacturing Company, 2025 Annual Report, https://investor.tsmc.com/static/annualReports/2025/english/index.html

[31] Amkor Technology, annual reports and filings, https://ir.amkor.com/financial-information/annual-reports-and-proxies

[32] ASE Technology Holding, annual reports, https://www.aseglobal.com/investors/annual-reports/

[33] National Institute of Standards and Technology, AI Risk Management Framework, https://www.nist.gov/itl/ai-risk-management-framework

[34] National Institute of Standards and Technology, AI standards engagement plan, https://www.nist.gov/artificial-intelligence

[35] US Department of Commerce, CHIPS for America, https://www.nist.gov/chips

[36] US Bureau of Industry and Security, export administration regulations, https://www.bis.gov/regulations/ear

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

[38] US Department of Justice, Guideline 6 on entrenching or extending a dominant position, https://www.justice.gov/atr/merger-guidelines/applying-merger-guidelines/guideline-6

[39] US Department of Justice, Guideline 9 on multi-sided platforms, https://www.justice.gov/atr/merger-guidelines/applying-merger-guidelines/guideline-9

[40] Federal Trade Commission, Hart-Scott-Rodino premerger notification programme, https://www.ftc.gov/enforcement/premerger-notification-program

[41] European Commission, horizontal merger guidelines, https://competition-policy.ec.europa.eu/mergers/legislation_en

[42] European Commission, Digital Markets Act, https://digital-markets-act.ec.europa.eu/

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

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

[45] IFRS Foundation, IAS 36 Impairment of Assets, https://www.ifrs.org/issued-standards/list-of-standards/ias-36-impairment-of-assets/

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

[47] Financial Accounting Standards Board, Topic 805 Business Combinations, https://asc.fasb.org/topic&trid=2125580

[48] World Intellectual Property Organization, IP valuation, https://www.wipo.int/sme/en/value_ip_assets/

[49] OECD, competition in the digital economy, https://www.oecd.org/competition/

[50] MLCommons, benchmark working groups and governance, https://mlcommons.org/working-groups/

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

This paper is part of a continuing series on the structure of private and alternative markets. The views expressed are the author's own. The paper is for information only, describes market structure in general terms, and does not constitute investment, legal, tax or regulatory advice or a recommendation in respect of any security, vehicle or counterparty.