Compute Sovereignty

How Geopolitics, Admission, and Integration Change Architecture, Ownership, and Silicon Strategy

Dana Xiadani
Founder & Chief Architect, Living Cipher
dana@livingcipher.com | www.livingcipher.com
Living Cipher Analysis 005 | September 2026 | Version 1.0
DOI: 10.5281/zenodo.22288858

Central claim

Execution, hosting, fabrication, data authority, and NRE do not by themselves determine ownership of reusable computation. In heterogeneous and transnational compute, geopolitical and procurement constraints can become deployment requirements; those requirements shape admission architecture; admission architecture determines who controls the right to participate; and those control rights affect value capture. In heterogeneous systems, integration itself is an admission event: the package or system integrator may have to decide whether a supplied die can be composed into a system for which the integrator will assume lifecycle, qualification, security, and customer obligations. A component can therefore be functionally correct yet economically inadmissible before the target datacenter ever evaluates it. When missing identity, state, provenance, lifecycle, authorization, or containment capabilities are deferred to the integrator, remediation cost can rise and architectural control can migrate up the stack.

Abstract

Semiconductor strategy increasingly distributes computation across chiplets, accelerators, datacenters, foundries, device partners, and sovereign deployment environments. Treating execution, infrastructure ownership, fabrication, data authority, engineering spend, legal title, and effective control as one concept creates a category error. This analysis separates three ledgers - cost, control, and value - and defines compute sovereignty as retaining sufficient architectural, lifecycle, and rights control to instantiate, authorize, modify, port, recover, and economically exploit reusable computation without another party becoming an unavoidable control point. It traces a causal chain from geopolitical and procurement constraints to deployment requirements, admission architecture, remediation ownership, control rights, value capture, and capital outcome. In heterogeneous systems, admission is staged: a package or system integrator may reject or remediate a component before the platform/OEM, deployment infrastructure, or procurement authority evaluates it. The economic question is therefore not only whether a component can be admitted, but who must solve missing architecture, who capitalizes that solution, and whether doing so changes the control and value assumptions underlying the financing. An illustrative model applies conventional discounted-cash-flow, expected-value, conditional-probability, and scenario-loss logic to deployment admission. It shows how late remediation, requalification, schedule delay, downstream rejection risk, and transferred control can alter the value of otherwise functional silicon. The paper does not argue that every chiplet needs the same trust mechanism or that system-level admission necessarily transfers supplier control. It argues that target deployment, the minimum component contract, and the allocation of canonical implementation, lifecycle, authorization, substitution, and derivative rights must be explicit before architecture becomes expensive to change.

Keywords: compute sovereignty; chiplets; semiconductor architecture; deployment admissibility; market access; hardware trust; provenance; attestation; systems integration; foundry strategy; capital allocation; strategic rights; technical diligence

1. The Ownership Question Is Being Asked at the Wrong Layer

A workload can execute on someone else's server without transferring ownership of the reusable computation. A foundry can manufacture a die without owning the computer embodied by the design. A ministry can control patient-data policy without owning the model, accelerator, or hardware IP. A chiplet integrator can own the package while individual suppliers retain their underlying technology. None of these statements is paradoxical. The confusion begins when execution, possession, manufacture, funding, and control are allowed to collapse into one word: ownership.

Analysis 003 separated data sovereignty, compute sovereignty, and economic sovereignty in infrastructure programs.[2] Analysis 004 then traced material technical premises into capital commitments and release decisions.[3] The present paper asks a narrower question across semiconductor and infrastructure boundaries: which party controls the reusable computer when the functions required to execute, admit, update, recover, manufacture, and evolve that computer are distributed among several organizations?

The question matters because modern heterogeneous systems can assign different pieces of the value chain to different parties. One party may invent the compute. Another may fabricate it. A third may integrate it into a package. A fourth may own the datacenter. A fifth may define the admission policy. A sixth may control the data. A seventh may own the customer relationship. Architecture determines whether these roles remain separable or silently become control rights over the computer itself.

Working definition. In this analysis, compute sovereignty means retaining sufficient architectural, lifecycle, and rights control to instantiate, authorize, modify, port, recover, and economically exploit reusable computation without another party becoming an unavoidable control point. Legal title and effective control are distinct: where this paper discusses ownership as architectural or economic control rather than IP title, the distinction is intentional.

The causal chain used throughout is equally specific: geopolitical or procurement constraints change deployment requirements; deployment requirements shape admission architecture; admission architecture allocates control rights; control rights affect value capture; and those effects change the capital outcome. Geopolitics does not directly determine a circuit. It changes the feasible architecture and the value of preserving particular control boundaries.

2. Three Ledgers: Cost, Control, and Value

A useful ownership review should keep at least three ledgers separate. The cost ledger asks who paid to create and integrate the capability. The control ledger asks who can authorize, modify, update, revoke, reproduce, port, or independently evolve it. The value ledger asks who captures product margin, compute margin, licensing income, derivative rights, data advantage, learning, and future option value. The ledgers can align, but they do not have to.

Exhibit 1. The three-ledger ownership test

Ledger	Primary question	Typical evidence
Cost	Who funded the work and which costs were partner-conditioned?	Engineering labor, EDA/IP, verification, qualification, package/test, integration NRE, external services
Control	Who can authorize, inspect, modify, update, revoke, port, reproduce, or continue the capability?	Keys, signing authority, source/RTL access, lifecycle authority, admission policy, derivative rights, portability
Value	Who captures the durable economic upside if the capability succeeds?	Product/compute margin, royalties, manufacturing economics, derivative silicon, platform option value, customer control

This separation prevents a familiar error: assuming that the party carrying the NRE therefore owns the reusable computation, or assuming that the party hosting the workload therefore owns the computer. Either may be true under a particular agreement. Neither follows from the technical fact alone.

3. Geopolitics Changes the Feasible Architecture

Geopolitics matters when it changes what can be purchased, deployed, trusted, exported, financed, or admitted. It should not be reduced to slogans about national origin. The relevant variables are concrete: jurisdiction, foreign ownership or influence, provenance, supply-chain tiers, key custody, operating authority, export access, sourcing concentration, and the ability to replace a provider without redesigning the system.

NIST SP 1326 makes several of these variables explicit in ICT supplier due diligence. Its five components are foreign ownership, control, or influence; provenance; resilience; foundational cyber practices; and supply-chain tiers.[5] These are acquisition and risk-assessment dimensions, but their consequences can propagate backward into architecture when a supplier, component, hosting environment, or manufacturing route would otherwise become difficult to replace.

This is the bridge between compute sovereignty and semiconductor strategy. A technically superior component can become economically inferior if it narrows the accessible customer set, creates an unacceptable jurisdictional dependency, or requires a partner to own the control plane that the supplier intended to monetize.

This is not a claim that jurisdiction or national origin automatically determines trustworthiness. The architecture consequence appears when concrete acquisition, export, provenance, key-custody, operating, or sourcing requirements alter the feasible supplier set, integration boundary, or replacement path. Those constraints then propagate into deployment and admission requirements.

For diligence, the relevant geopolitical/manufacturing configuration can be represented as G = {foundry ownership, fabrication jurisdiction, packaging/test jurisdiction, supply-chain provenance, foreign ownership or influence, export/re-export conditions, key or operating authority, and alternate-source radius}. The same compute architecture can therefore have different deployment economics under different G even when its RTL and functional behavior are unchanged.

Geopolitics enters the capital model through conditional admissibility and recovery cost, not through a universal country score. A manufacturing path is economically relevant when it changes which relying parties can procure, admit, support, or replace the resulting component on acceptable terms.

4. Interoperability Is Not Admission

Chiplet standards correctly focus on making heterogeneous dies composable. UCIe 3.0 increases bandwidth and extends manageability, including early firmware download, priority sideband signaling, throttle and emergency-shutdown mechanisms.

[6] These are important system capabilities. They do not, by themselves, answer every relying-party question about whether a particular die should be trusted or admitted.

The distinction is simple: interoperability establishes that components can communicate according to an interface. Admission establishes that the system is willing to let a particular component participate under a particular policy.

Open Compute Project work makes this distinction concrete. A proposed CDX chiplet model includes an optional security agent for trusted supply-chain traceability; it describes authenticating a chiplet to a system-level root of trust and withholding data transfers when authentication fails.[8] The February 2026 Foundation Chiplet System Architecture goes further by defining trusted security agents, security-platform boot state, platform attestation, hardware-security lifecycle state, and a primary/secondary trusted-security-agent model across chiplets.[7]

DMTF's SPDM family similarly standardizes component authentication, attestation, key exchange, secured messaging, and authorization.[9] Those mechanisms can carry evidence and protect communication; the relying system still determines whether that evidence satisfies its admission policy.

The standards landscape will continue to evolve, and not every product will implement the same mechanism. The capital implication does not depend on one particular protocol. A supplier must know what evidence the target integrator, datacenter, sovereign environment, or regulated customer will require — and whether the design can expose and enforce that evidence before the silicon is difficult to change.

Exhibit 2. Four layers from interoperability to market access

Layer	Primary question	Typical evidence or consequence
Interoperability	Can the components communicate and compose?	Electrical/protocol compatibility, link training, data exchange, interface conformance
Platform admission	Will this system authorize this particular component to participate?	Identity, measured state, attestation, authorization, gating, containment
Deployment admissibility	Will the target OEM, datacenter, sovereign, defense, or regulated environment accept the architecture?	Required trust evidence, lifecycle controls, provenance, operational and supply-chain requirements
Market access	Can the implementation be legally, contractually, and commercially procured and deployed?	Export, procurement, jurisdiction, FOCI, sourcing, qualification, and policy constraints

These layers are coupled but not interchangeable. A design can pass interface compliance and still fail platform admission; it can pass platform admission in one system and still fail deployment or procurement requirements in another. The diligence task is to identify which layer underwrites the investment case and what evidence must exist before irreversibility increases.

5. Integration Is the First Admission Gate

In heterogeneous compute, the datacenter is not necessarily the first relying party. The package or system integrator must decide whether a supplied die can be composed into a system for which the integrator will assume performance, lifecycle, security, qualification, and customer obligations. Integration is therefore itself an admission event.

A chiplet can be electrically interoperable and still fail integrator admission if its identity, measured state, provenance, lifecycle, authorization, or failure behavior cannot be reconciled with the composed system. The integrator may reject the component, require supplier remediation, constrain it behind an external controller or trust boundary, or accept it under a narrower operating model. The Foundation Chiplet System Architecture primary/secondary trusted-security-agent model and the CDX example of withholding data transfer after failed authentication illustrate that admission and lifecycle decisions can be enforced inside a composed system, before the finished platform reaches a datacenter.[7][8]

An integrator should not be modeled as a costless absorber of deferred architecture. If it must build enrollment, policy, gating, attestation aggregation, update, quarantine, or recovery around a frozen die, it funds NRE, absorbs schedule and qualification risk, and may become responsible for satisfying the next relying party. The economically relevant question is therefore not only: Can the chiplet be admitted? It is: At each admission gate, who must solve the missing architecture, who capitalizes that solution, and what control rights are created as a consequence?

Here, capitalizes means the party that funds or internalizes the engineering, schedule, qualification, inventory, support, or customer-risk burden required to make the component acceptable. That party need not receive ownership of the supplier's RTL or patents. But the remediation can create durable authority over enrollment, lifecycle, routing, containment, substitution, or the customer-facing trust contract.

Admission authority is therefore a separate diligence object from asset ownership. The supplier may own the die while the integrator owns the authority that enrolls the component, evaluates its approved state and composition evidence, authorizes producer-consumer relationships, and enforces refusal, quarantine, or revocation. Where that authority is reusable across multiple suppliers, it can become an integrator platform asset rather than a one-time remediation.

Exhibit 3. The admission-consequence test

Admission gate	What may be missing	Who may capitalize the solution	Control / economic consequence
Integrator admission	Component identity, measured state, provenance, lifecycle, authorization, containment, or a usable evidence interface	Chiplet supplier, package/system integrator, or both	Design-win refusal; supplier-funded remediation; integrator-owned wrapper, enrollment, gating, or substitution boundary
Platform / OEM admission	Composed-system state, supportability, qualification evidence, update/recovery path, or platform-facing attestation	Integrator, OEM/platform owner, supplier, or shared program	Additional qualification and schedule exposure; platform controller and lifecycle authority may move up-stack
Deployment / datacenter admission	Infrastructure-specific evidence, authorization, provenance, containment, operational policy, or incident-recovery behavior	Integrator, platform operator, supplier, or customer program	Execution authorization, quarantine policy, customer-visible trust relationship, or constrained deployment scope
Market / procurement admission	Jurisdiction, export, FOCI, sourcing, supply-chain, provenance, or acquisition requirements	Supplier, integrator/OEM, or procurement program	Alternate sourcing, requalification, market exclusion, route dependency, or changed replacement radius

These gates are sequential in value terms even when engineering work overlaps. Failure at an early gate can truncate downstream market value before a datacenter or end customer ever evaluates the component. Acceptance can also change the value thesis: if another party capitalizes the missing layer, diligence must determine whether the resulting authority is merely implementation support or a durable control point over participation, lifecycle, substitution, or the customer relationship.

6. Trust Debt Can Become Stranded Capital

Analysis 001 argued that an accelerator that cannot establish cryptographic identity, attest its state, detect integrity failure, and participate in containment may be operationally inadmissible to the infrastructure it was built to serve.[1] The new question is what happens economically when that discovery occurs late.

A component can meet throughput, latency, power, yield, thermal, and interface goals and still fail a customer's admission architecture. The result is not necessarily total commercial failure. Lower-assurance markets may remain available. The risk is that the highest-value deployments — hyperscale, sovereign, defense, regulated, or critical infrastructure — may require evidence or lifecycle properties that the original die cannot provide without material remediation.

Exhibit 4. How late trust requirements can change the capital outcome

Late discovery	Likely remediation	Capital consequence
No durable device identity or enrollment path	External security controller, board/package change, or silicon respin	New NRE, schedule slip, additional BOM/package burden
No trustworthy measured-state or attestation path	Firmware redesign, root-of-trust integration, measurement architecture change	Re-verification, lifecycle redesign, possible silicon/FPGA change
No enforceable admission/containment path	Integrator-owned wrapper, host/BMC policy, DMA or fabric gating added outside supplier boundary	Supplier may lose control of execution policy and become more substitutable
No update/recovery/revocation architecture	Operational workaround or redesign of lifecycle mechanisms	Qualification friction, reduced deployment scope, support liability
Provenance/FOCI requirements discovered after sourcing is fixed	Supplier/package/foundry substitution or market exclusion	Requalification, inventory/capacity exposure, market-access loss

The key phrase is target-market admissibility. Security requirements vary by use case. The diligence failure is not choosing the wrong universal security feature. It is failing to identify the trust and evidence requirements of the economically relevant deployment before capital is committed.

Capital translation. Analysis 001 placed redesign, stranded inventory, capacity commitments, requalification, and other failures of underlying assumptions inside an expected-loss term, E[L].[1] Admission failure is one such assumption failure, but it has an additional feature: the silicon can remain technically correct while the economically important deployment becomes unavailable.

For admission risk, the economic exposure therefore includes more than remediation NRE. It includes the probability that the target market or platform will admit the component, the timing and cost of recovering admissibility, and any durable value surrendered if another party must own the admission/control layer.

The next section makes that translation explicit with an illustrative model and concrete failure paths. The model is intended to expose decision structure and sensitivity, not to provide an accounting identity or a market valuation standard.

Cryptographic modernity is not deployment admissibility. Post-quantum cryptography can protect the authenticity and integrity of a signed statement, but it does not create the identity root, enrollment process, measured state, lifecycle authority, authorization policy, quarantine behavior, or recovery path required to decide whether a component should participate. NIST IR 8615 is useful because it frames next-generation hardware security as a lifecycle and ecosystem problem involving interoperable trust models, provenance, cryptographic identities, attestation, verification, lifecycle-aware controls, and procurement incentives.[4] The architecture question is therefore how cryptography, identity, evidence, lifecycle, and admission compose into a system whose claims can be evaluated and enforced by the relying party.

7. What Stranded Capital Looks Like: A Chained Admission Model

Analysis 001 compared irreversible silicon with the best less-irreversible alternative using risk-adjusted NPV and an expected-loss term for redesign, stranded inventory, capacity commitments, requalification, and other failures of the assumptions embedded in hardware.[1] Analysis 005 extends that logic to deployment admission. The relevant question is not simply what a missing trust or provenance capability costs to add. It is how that omission changes the probability, timing, control boundary, and economically accessible market of the resulting silicon.

Series model inheritance. The model does not begin from zero. Analysis 001 supplies the fixed-versus-flexible NPV test, expected-loss term E[L], irreversibility, and replacement-radius logic.[1] Analysis 002 supplies the sequence claim -> required evidence -> trust boundary -> partition -> substrate, treating required observation and machine evidence as architecture and admission constraints.[10] Analysis 003 extends the frame to dependency, rights transfer, portability, and economic sovereignty.[2] Analysis 004 carries those outputs into premise-to-commitment and recoverability decisions while preserving the inherited valuation rules and anti-double-counting discipline.[3] Analysis 005 adds one new unit of analysis: admission as conditional economic access through sequential relying-party gates, with remediation ownership traced into capital exposure and control/value consequences.

Model derivation and interpretation. The equations below are not proposed as new financial mathematics, accounting identities, or empirical laws. They begin with conventional discounted-cash-flow value for deployment m. If access to that deployment depends on successful admission event A_m under geopolitical/manufacturing configuration G, expected accessible value is P(A_m | G) times V_m. Where admission occurs through sequential relying-party gates, P(A_m | G) is expanded using the chain rule of probability; no independence assumption is made. Direct late-recovery costs are represented as scenario-weighted expected losses, while transferred control is modeled separately as a counterfactual option-value term.

The model is deliberately conservative about double counting. Each economic consequence appears once: a launch delay is represented either by shifting the cash-flow timing inside V_m or by a separate schedule-loss term, but not both; lost market access is represented through P(A_m | G) times V_m rather than subtracted again as remediation. Admission probabilities are diligence inputs, not constants. They should be supported by integrator requirements, qualification criteria, procurement rules, prior program evidence, or ranges and sensitivity analysis where point precision is not justified. The control-value term should be quantified only when evidence supports a counterfactual difference between supplier value with and without the transferred authority; otherwise it should remain a range or qualitative sensitivity.

$$V_{DA}(G) = \Sigma_m \, P(A_m \mid G) \, V_m - C_{admission}(G) - E[L_{admission} \mid G] - \Delta V_{control}(G)$$

where V_DA(G) is deployment-adjusted value under geopolitical/manufacturing configuration G; P(A_m | G) is the probability that target market or deployment m admits the component under the proposed architecture and configuration; V_m is the discounted economic value of that deployment if admitted; C_admission(G) is the incremental cost of the admission architecture intentionally designed into the program; E[L_admission | G] is expected direct loss from late remediation, requalification, stranded inventory or commitments, and other recovery actions; and ΔV_control(G) is the durable option value surrendered if remediation transfers authority that the supplier intended to retain.

$$P(A_m \mid G) = P(A_I \mid G) \times P(A_P \mid A_I, G) \times P(A_D \mid A_I, A_P, G) \times P(A_{MKT} \mid A_I, A_P, A_D, G)$$

where A_I is integrator admission; A_P is platform/OEM admission; A_D is deployment or infrastructure admission; A_MKT is market/procurement admission; and G is the geopolitical/manufacturing configuration. This is a conditional chain, not an independence assumption. Failure at an earlier gate prevents the value associated with later gates from being realized, even if the later relying party would otherwise accept the technology. Where a particular deployment does not require one of these gates, that factor is omitted or treated as satisfied with probability 1.

$$E[L_{admission} \mid G] = \Sigma_s \, p_s(G) \, (C_{remediation,s} + C_{requalification,s} + C_{schedule,s} + C_{inventory,s})$$

This decomposition keeps planned admission architecture, contingent recovery cost, and transferred control analytically separate. Lost access to a target market is represented through P(A_m | G) times V_m. Direct recovery costs belong in E[L_admission | G]. A product can ship successfully and still surrender strategic option value, which is why ΔV_control(G) remains a separate counterfactual term rather than a direct remediation cost.

Illustrative case. Assume a four-year target deployment capable of producing $18 million per year of contribution cash flow if admitted, discounted at 12 percent. Its unadjusted present value is approximately $54.7 million. The figures below are hypothetical and are not estimates for a specific product, customer, foundry, integrator, or market.

Exhibit 5. Illustrative deployment-admission value gap

Architecture path	Illustrative assumptions	Deployment-adjusted value	Gap vs. early path
Admission designed in before irreversibility increases	P(admission)=85%; no launch delay; $1.2M incremental admission-architecture NRE; supplier retains control plane	$45.3M	-
Late external workaround after architecture is fixed	P(admission after workaround)=75%; 9-month launch delay; $9M remediation + requalification; $10M transferred-control option value	$18.7M	-$26.6M
No acceptable admission path for the target deployment	P(admission)=25%; no remediation undertaken; silicon remains functional but economically relevant market access is severely constrained	$13.7M	-$31.6M

The arithmetic is intentionally simple. The early path begins with the approximately $54.7 million present value, applies an 85 percent admission probability, and subtracts $1.2 million of deliberate admission-architecture NRE. The late-workaround path shifts the same cash-flow stream nine months, applies a 75 percent post-remediation admission probability, and subtracts $9 million of direct remediation/requalification plus $10 million of surrendered control value. The resulting approximately $26.6 million gap is not “the cost of security.” It is the combined capital consequence of discovering a deployment requirement after the architecture has become expensive to change.

Integrator capital at risk. The late-workaround case also creates a second ledger that belongs to the integrator rather than the supplier. If the integrator accepts a frozen component with unresolved admission architecture, its own exposure can include package/interposer work, external control logic, firmware, qualification, inventory, schedule, customer commitments, and the expected consequence of a later platform or deployment rejection.

$$E[L_{integrator}] = \Sigma_s\, p_s\, (C_{package,s} + C_{wrapper,s} + C_{qualification,s} + C_{schedule,s} + C_{inventory,s} + C_{customer,s})$$

That exposure explains why an integrator has a rational basis to reject the component, require supplier-funded remediation, economically discount it, or build a reusable admission layer under its own control. The supplier cannot assume that deferred architecture will be absorbed without changing transaction economics or bargaining power.

Three concrete failure paths make the model easier to interpret:

- **Electrically compatible, admission-incompatible chiplet.** A die meets its interface, throughput, power, and thermal targets, but the target datacenter requires durable component identity, measured state, lifecycle evidence, and an enforceable failure response. If the die cannot expose that evidence, the recovery choices are a respin, a board/package security controller, a narrower market, or a different integration boundary. The stranded value includes whichever high-value deployments can no longer be reached on the original schedule.
- **Integrator saves the shipment and captures the control point.** An external controller or platform root of trust makes the component admissible without a silicon respin. Unit sales survive, but enrollment, workload authorization, update, quarantine, revocation, and the customer-visible trust contract now belong to the integrator. The loss is therefore not only remediation cost; it can also appear as reduced pricing power, higher substitutability, and surrendered derivative-platform option value.
- **Geopolitical or provenance constraint changes the feasible route.** A foundry, package, supplier, or hosting path that was technically adequate fails a target buyer's provenance, foreign-ownership/influence, export, or supply-chain requirement after sourcing and qualification have narrowed. The transistor does not fail. The accessible deployment set changes. Alternate sourcing can then create requalification cost, inventory or capacity exposure, schedule delay, and a larger replacement radius.
- Architecture passes component diligence but fails consequence diligence. RTL, PPA, verification, interface compliance, and the foundry plan can all be credible while the investment thesis remains wrong because nobody traced who must supply the missing admission layer, whose capital is exposed, or what authority that remediation creates. Nothing in the component fails; the diligence boundary was simply drawn too narrowly.

The decision implication is the same as in Analysis 001: the evidentiary burden should rise with irreversibility. In 005, however, the evidence must cover not only whether the computation deserves fixed implementation, but whether the target deployment can admit that implementation and whether the chosen admission architecture preserves the control rights on which the economic thesis depends.

8. When the Integrator Builds the Missing Admission Layer

A large integrator may rationally absorb or remediate a supplier's missing admission architecture. It can place the component behind a management controller, host policy, platform root of trust, external security device, or package-level control plane. That may be the correct system partition. The economic issue is that the integrator is capitalizing engineering, schedule, qualification, and potentially downstream customer risk that the supplier deferred; the supplier must therefore understand what consideration, authority, or bargaining leverage accompanies that investment.

Deferred architecture therefore has a transfer price. The consideration may appear as lower component economics, supplier-funded engineering, stronger lifecycle and support obligations, integrator-owned management infrastructure, or other transaction-specific terms. The paper does not assume a particular commercial remedy; it identifies the underlying economic fact that the party assuming capital and customer risk has an incentive to capture value or control in return.

System admission authority does not automatically transfer supplier strategic control. A supplier can remain strategically sovereign while a platform owns system admission if the supplier retains the canonical implementation of its differentiated capability, appropriate lifecycle and version authority over that capability, derivative and porting rights, and an integration contract that does not make the platform independently able to recreate the supplier's reusable computer. The architectural question is therefore which authorities are required for system operation and which authorities are essential to the supplier's economic differentiation.

Control transfer becomes economically consequential when system admission is coupled with sufficient authority or artifacts to control the canonical implementation, supplier-specific lifecycle, substitution boundary, derivative rights, or customer-facing authorization in a way that erodes the supplier's differentiation. Unit sales can survive while pricing power, switching costs, customer ownership, derivative-platform option value, or acquisition value migrate elsewhere. The diligence problem is therefore not whether revenue remains possible; it is whether the architecture still supports the kind of revenue and enterprise value assumed by the financing thesis.

Exhibit 6. Functional ownership versus system control

Supplier retains	Integrator may control if admission is externalized	**Economic effect**
Algorithm / datapath / die IP	Enrollment and identity policy	Supplier owns computation but may not own admission
Performance differentiation	Workload authorization and DMA/fabric access	Execution may depend on integrator policy
Manufactured component	Update, quarantine, recovery, revocation	Lifecycle relationship may shift up-stack
RTL or netlist ownership	Platform-facing attestation and provenance contract	Integrator may own the trust interface seen by the customer
Unit sales	System composition and replacement authority	Supplier may become more substitutable

The conclusion is not that every supplier must own the entire system trust plane. It is that externalizing admission is an architecture and value-capture decision, not an implementation afterthought, and that downstream system authority should not be confused automatically with ownership of the supplier's reusable computer.

Financing can precede admission. Early-stage or strategic capital may rationally fund unresolved architecture, integration, qualification, and market-access risks. It may also be technically appropriate for identity, lifecycle, admission, or containment functions to reside at the customer or platform layer rather than inside every component. The diligence question is therefore not whether an unresolved admission requirement makes a company unfinanceable. It is whether the frozen architecture preserves a feasible minimum contract for satisfying that requirement before irreversibility, which party must capitalize that path, and whether doing so changes the control and value assumptions underlying the financing. 'The customer platform will handle it' is an architecture assumption, not the absence of one.

9. Manufacturing Dependency Is Not Ownership - But It Can Become Control

Manufacturing is another boundary that is often confused with ownership. A foundry necessarily receives enough design information to fabricate the contracted device. That fact does not inherently confer independent rights to reproduce the design for other customers, modify the computer, control its roadmap, or capture its derivative economics. Those are rights questions layered on top of manufacturing.

The same distinction applies to chiplet assembly and advanced packaging. A package integrator may own the interposer, substrate, thermal solution, assembly flow, system controller, or even the primary security agent. Depending on the architecture and agreement, those positions may or may not translate into strategic control of the composed computer. The diligence task is to state the boundary explicitly before physical integration makes it difficult to renegotiate.

Manufacturing dependency becomes effective control when exit is economically or technically prohibitive. The relevant tests include process-node portability, foundry-specific hard IP, mask and tooling economics, qualification time and cost,

package/interposer dependencies, advanced-packaging capacity, export access, jurisdictional restrictions, alternate-source readiness, and the radius of redesign required to replace the provider. None of these automatically transfers legal ownership. Together, however, they can determine whether the supplier retains a practical right to continue the computer on acceptable terms.

A useful test is survivability: if the original compute supplier disappeared tomorrow, which parties could continue to instantiate, modify, port, update, and evolve the capability? The farther a partner can continue independently, the closer the relationship is to reusable implementation transfer rather than ordinary integration.

10. Admission Gates Belong Before Tapeout

The ordinary pre-silicon review should therefore include staged admission gates alongside performance, power, area, manufacturability, and economics. The review should begin with integrator admission and continue through the platform/OEM, deployment/infrastructure, and market/procurement gates that materially underwrite the investment case. Each gate is target-specific. It asks whether the implementation can produce and enforce the required evidence, who must supply any missing architecture, and whether the resulting control boundary preserves the economic thesis.

Exhibit 7. Pre-silicon staged admission review

Question	Evidence before increasing irreversibility
Who must trust the component?	Named integrators, datacenters, OEMs, governments, regulated environments, or package owners
What must they establish?	Identity, provenance, approved state, firmware/configuration, lifecycle, authorization, supply-chain facts
How is evidence exposed?	Defined management/attestation path, responder, certificate/measurement mechanism, sideband or host-accessible interface as appropriate
What is enforced on failure?	Refuse admission, withhold data transfer, disable compute/DMA, quarantine, degrade, recover, revoke
Who owns lifecycle control?	Keys, signing, update authority, re-enrollment, revocation, incident recovery, end-of-life
What happens if requirements change?	Programmable security boundary, reserved hooks, versioned interfaces, bounded replacement radius
What value is lost if the integrator supplies the missing layer?	Pricing power, platform control, derivative rights, customer relationship, strategic option value

A design can clear this gate by implementing the required trust capabilities internally, by composing with a trusted system controller, or by another architecture that the target environment accepts. The decision is architectural, not ideological. What matters is that the path is known before the program assumes the resulting silicon is deployable.

11. Capital Allocators Should Ask Who Owns the Right to Execute

The capital meeting should not stop at “Does the chip work?” or even “Can it be manufactured?” It should trace the component through each admission gate and ask what must remain true for the original value thesis to survive. The central diligence question is not only whether the chiplet can be admitted, but who bears the consequences of making it admissible.

Why conventional technical diligence can miss the problem. Component diligence commonly proves that the RTL, PPA, verification, interfaces, security primitives, foundry route, and implementation plan are credible. Commercial, legal, security, and supply-chain reviews may also each be individually sound. The blind spot appears when no review traces those answers through the composed system and asks who must remediate what remains, whose capital becomes exposed, and whether the remediation changes control or value capture. A technically credible chip can therefore support an economically incorrect investment thesis.

Architecture-consequence diligence asks a different question: if another party must supply what remains to be engineered, what does that party consequently fund, control, or become able to substitute? This connects a technical gap to its remediation owner, capital exposure, control rights, and residual value rather than treating integration as an undifferentiated downstream task.

- At each admission gate - integrator, platform/OEM, deployment/infrastructure, and market/procurement - who must solve the missing architecture, who capitalizes that solution, and what control rights are created as a consequence?
- If an integrator accepts the component, what package, wrapper, qualification, inventory, schedule, support, or downstream-admission capital does it put at risk - and what economic or control consideration would rationally accompany that exposure?
- Which customers or infrastructure environments materially underwrite the volume forecast, and what trust evidence do they require?
- Can the device or chiplet establish identity, state, provenance, and lifecycle evidence in a form the relying system can consume?

- If a required capability is missing, what remediation is required, which party will fund and implement it, and does that party acquire durable authority over enrollment, lifecycle, execution policy, substitution, or the customer-facing trust interface?
- Does the proposed integration leave the supplier with a strategic compute platform or reduce it to a replaceable component behind the integrator's admission plane?
- If the customer or platform owns admission, what minimum identity, state, lifecycle, evidence, and enforcement contract must the frozen component preserve for that assumption to remain valid across the markets used in the forecast?
- Is the financing thesis underwriting merchant-component economics or platform economics - and does the post-integration control structure support the margins, switching costs, customer relationship, derivative rights, and option value implied by that thesis?
- Which rights must cross the boundary for manufacture, integration, or hosting — and which rights do not?
- Could a supplier, foundry, package integrator, datacenter, or jurisdiction change make the current architecture commercially inadmissible?
- What is the kill condition: what evidence would cause management to defer tapeout, reserve a trust hook, change the partition, or retain programmability?

12. Decision Rule: Preserve the Computer Until Transfer Is Worth More Than Ownership

Compute sovereignty is not autarky. A company does not need to own its foundry, datacenter, package line, sensor, network, or every security function in order to retain a coherent computer. It needs an architecture and rights model that preserve the capabilities it intends to monetize and the option to evolve them.

The default rule is therefore to use the smallest technical and rights transfer that satisfies the partner's legitimate need. A supplier can preserve strategic control by exposing a stable, versioned capability contract while allowing a downstream platform to own its own system-admission boundary. An API, signed runtime, supplied accelerator, bounded hard-IP block, manufacturable design package, and reusable core implementation are not interchangeable. Each transfers a different amount of implementation knowledge, lifecycle authority, substitution power, derivative rights, and future option value.

The rights-transfer trigger is crossed when another party receives three things together: durable rights or artifacts sufficient to continue the capability; practical independence from the original supplier; and material future option value from independently instantiating, modifying, porting, updating, or evolving a consequential portion of the reusable computer. Routine maintenance rights that do not create practical independence are not equivalent. Once all three conditions are present, the economics should be evaluated as strategic implementation transfer rather than routine integration.

Conclusion: Interoperability Lets the Chiplet Speak. Admission Lets It Participate.

Heterogeneous compute is making the physical computer more distributed at the same time that infrastructure, regulation, procurement, export constraints, and supply chains are making deployment requirements more consequential. The resulting capital chain is straightforward: external constraints change deployment requirements; deployment requirements shape admission architecture; admission architecture allocates control; and control affects value capture. Admission is itself a chain: an integrator can refuse a component before an OEM, datacenter, government buyer, or other downstream relying party ever evaluates it. The geopolitical/manufacturing configuration changes the probability and cost of clearing those gates without implying that jurisdiction alone determines trustworthiness.

A chiplet can be electrically interoperable yet commercially inadmissible. It can also be admitted by an integrator only after that integrator supplies missing enrollment, lifecycle, evidence, or containment architecture. That does not automatically extinguish supplier control: a downstream platform may own system admission while the supplier retains the canonical implementation, version/lifecycle authority over its capability, derivative rights, and a stable integration contract. Control migrates when remediation also gives the downstream party consequential authority over the supplier's differentiated implementation, substitution boundary, derivative path, or customer-facing authorization. An accelerator can be cryptographically modern yet lack the identity, measurement, lifecycle, and policy hooks a target platform needs. A foundry can fabricate the design without owning the computer. A datacenter can execute the workload without owning the reusable compute. A country can control its data while still surrendering the economics of the compute layer. The question is not whether one party performs system admission, but whether the resulting partition preserves the control and value thesis the supplier and its investors intended to retain.

The architecture task is therefore not merely to distribute computation. It is to decide where identity, admission, lifecycle, manufacturing dependency, rights, and economic control should reside - and to prove that the chosen boundaries remain deployable and replaceable before the silicon becomes expensive to change.

The final capital question is not only what deserves to become hardware. It is: who will own the right to let that hardware participate?

Selected References

1. Dana Xiadani, “From Signal to Substrate: Architecture, Irreversibility, and the Capital Economics of Custom Silicon,” Living Cipher Analysis 001, Version 1.0, 2026. DOI: 10.5281/zenodo.21958390.

2. Dana Xiadani, “Sovereign Clinical Compute: Capital Formation, Computational Sovereignty, and the Economics of Infrastructure Control,” Living Cipher Analysis 003, Version 1.0, 2026. DOI: 10.5281/zenodo.22021064.

3. Dana Xiadani, “Architecture Is Capital Allocation: Why Semiconductor Investment Decisions Begin Before the Capital Meeting,” Living Cipher Analysis 004, Version 1.0, 2026. DOI: 10.5281/zenodo.22150344.

4. S. Rekhi, N. Goren, and Y. Guo, “Workshop on Rolling Next-Generation Secure Hardware Into Standards,” NIST Internal Report 8615, September 1, 2026. DOI: 10.6028/NIST.IR.8615.

5. J. Boyens, R. McWhite, and L. Calloway, “NIST Cybersecurity Supply Chain Risk Management: Due Diligence Assessment Quick-Start Guide,” NIST Special Publication 1326, July 8, 2026. DOI: 10.6028/NIST.SP.1326.

6. UCIe Consortium, “UCIe 3.0 Specification,” released August 2025; specification overview and manageability highlights available at uciexpress.org/specifications.

7. Arm Ltd., “Foundation Chiplet System Architecture,” Version 1.0.0, Open Compute Project, February 10, 2026, §6.2, Trusted Security Agent.

8. A. Mastroianni et al., “Proposed Standardization of Chiplet Models for Heterogeneous Integration,” Open Compute Project, Open Domain-Specific Architecture (ODSA), Chiplet Design Exchange (CDX), “Optional: Security Agent,” p. 23.

9. DMTF, “Security Protocol and Data Model (SPDM) Specification,” DSP0274, Version 1.4.1, July 2, 2026; “Secured Messages Using SPDM Specification,” DSP0277, Version 2.0.0, December 8, 2025; “Security Protocol and Data Model (SPDM) Authorization Specification,” DSP0289, Version 1.0.0, December 8, 2025.

10. Dana Xiadani, “The Trusted Chip Is Not the Trusted Signal: How Signal Veracity Changes Hardware Partitioning in Consequential Systems,” Living Cipher Analysis 002, Version 1.0, 2026. DOI: 10.5281/zenodo.22019132.

Author Note

Dana Xiadani is Founder & Chief Architect of Living Cipher, where she works at the boundary of computation, architecture, trust, semiconductor strategy, and physical realization. Living Cipher Analysis examines decisions upstream of implementation and capital commitment: what computation deserves to become hardware, which assumptions are being capitalized, where technical choices become dependencies, and what evidence should exist before a system becomes expensive to change. This paper extends that framework to compute sovereignty, deployment admissibility, transnational semiconductor boundaries, control rights, and value capture. Its quantitative example applies conventional discounted-cash-flow, expected-value, conditional-probability, and scenario-loss logic to hypothetical inputs in order to expose decision structure and sensitivity; it is not a valuation standard or a forecast for any specific company, customer, market, or program. Proprietary Living Cipher scoring methods, implementation mechanisms, and protected architecture are not disclosed.

Suggested citation: Xiadani, Dana. 2026. “Compute Sovereignty: How Geopolitics, Admission, and Integration Change Architecture, Ownership, and Silicon Strategy.” Living Cipher Analysis 005, Version 1.0. DOI: 10.5281/zenodo.22288858.