Sovereign Clinical Compute

Capital Formation, Computational Sovereignty, and the Economics of Infrastructure Control

Dana Xiadani
Founder & Chief Architect, Living Cipher
dana@livingcipher.com
www.livingcipher.com

Living Cipher Analysis 003 | August 2026 | Version 1.0
DOI: 10.5281/zenodo.22021064

The Hidden Economics of Who Controls the Compute

Abstract

Artificial-intelligence infrastructure is often discussed as a capacity problem: more connectivity, more data centers, more accelerators, more cloud access. For clinically consequential systems in capital-constrained markets, that framing is incomplete. The financing instrument, ownership structure, operating authority, key custody, portability, derivative rights, local engineering capability, and compute location jointly determine who captures the economic value created by the system.

This paper extends the irreversibility framework developed in Living Cipher Analysis 001 from custom silicon to infrastructure.[7] It asks not simply whether capital can finance compute, but what becomes difficult to reverse when a particular form of capital finances a particular architecture. The analysis separates data sovereignty, compute sovereignty, and economic sovereignty; develops a risk-adjusted comparison of infrastructure-financing structures; connects policy and capital formation to bedside, edge, regional, and accelerator architecture; and establishes decision gates for determining when external capital accelerates useful deployment without transferring disproportionate control or future economic upside.

The central proposition is that capital architecture and compute architecture are the same decision viewed from opposite ends of the system. The relevant question is not merely who will finance the infrastructure. It is what the financing causes the architecture—and the future bargaining position—to become.

1. The Infrastructure Question We Keep Asking Too Late

Infrastructure programs are usually described as development, procurement, cloud, connectivity, or health-system problems until somebody has to decide who will finance them and under what conditions. Then apparently separate questions collapse into one decision: where will computation execute, who will own and operate the machines, who will control the keys and update path, which vendors become difficult to replace, who will acquire the engineering learning generated by deployment, and where the resulting economic value will accumulate.

This is particularly consequential in African AI infrastructure. The IMF's July 2026 analysis describes a region with large potential gains from AI but material constraints in electricity, connectivity, compute, skills, financing, and institutional capacity. It also documents significant new investment from Western, regional, and Chinese technology and infrastructure actors.[1] The World Bank similarly frames AI readiness around connectivity, compute, context, and competency, while warning that digital sovereignty becomes material when critical systems are built, owned, or operated by actors a country cannot fully control.[2][3]

The relevant question is therefore not whether foreign capital should participate. In many markets it must. The question is what the recipient is actually buying with the dependency it accepts.

Capital is not neutral once it determines architecture.

2. Architecture Is Where Financing Becomes Dependency

Living Cipher Analysis 001 argued that architecture determines where capital becomes irreversible.[7] The same logic applies one level above the chip. A financing structure can determine the vendor, operating model, cloud or data-center topology, procurement path, key-control regime, interfaces, update authority, and rights surrounding data or derivative improvements. Once hospitals, applications, models, operations, and staff become organized around those choices, switching is no longer a purchasing event. It becomes a redesign.

The evidentiary burden should therefore rise with the dependency radius of the financing structure.

Dependency radius is the portion of the operational, technical, contractual, and economic system that must be replaced, renegotiated, requalified, or rebuilt when a financing or infrastructure relationship changes.

Replacement radius, evidentiary locality, and dependency radius are related but distinct architectural quantities. Replacement radius measures how much of a physical implementation must change when an assumption fails. Evidentiary locality measures how far a claim-relevant property can travel before evidence for that property loses meaning.[8] Dependency radius measures how much of the technical, operational, contractual, and economic system must change when an infrastructure or capital relationship fails.

Replacement radius -> physical redesign consequence

Evidentiary locality -> evidence-placement constraint

Dependency radius -> infrastructure / capital failure consequence

Dependency radius is therefore descriptive until it is mapped to failure states and economic loss. For financing or infrastructure structure j, define:

$$D_j = \sum_s P(s \mid j) \times PV(C_{switch,s} + C_{redesign,s} + C_{requal,s} + C_{downtime,s})$$

where s indexes mutually distinguishable dependency events or scenarios; $P(s \mid j)$ is the probability of that event under structure j; and the cost terms represent switching, redesign, requalification, and downtime or service-continuity loss. If scenarios are not mutually exclusive, the analyst must use a scenario tree or conditional probabilities rather than simply summing overlapping expected losses.

A loan that finances interoperable equipment with locally controlled keys and documented exit interfaces has a different dependency radius from an integrated finance-build-operate arrangement in which the same provider controls infrastructure, management plane, proprietary interfaces, operational expertise, and upgrade path.

The headline cost of capital may be lower in the second case while the economic cost of exit is substantially higher.

$$V\_j = \sum\_t E[CF\_{clinical,j,t} + CF\_{commercial,j,t} + CF\_{capability,j,t} - C\_{infra,j,t} - C\_{dependency,j,t} - C\_{rights,j,t}] / (1+r\_j)^t + ROV\_j$$

where V_j is the value of financing/infrastructure structure j; CF_clinical is monetizable or decision-relevant clinical benefit when a defensible monetary proxy exists; CF_commercial is commercial free cash flow; CF_capability is cash-equivalent value of retained local or platform capability that is not already embedded in commercial cash flow; C_infra is financing, infrastructure, and operating cost; C_dependency is expected dependency loss; C_rights is economic value surrendered through contractual or control rights; r_j is the discount rate appropriate to the structure; and ROV_j is residual option value not already represented in forecast cash flows. Every effect appears once.

The model intentionally avoids adding a separate generic “project NPV” term to dependency, rights, and capability adjustments. Doing so would invite double counting because those effects may already be embedded in project cash flows.

where B_t is the economically capturable benefit created in period t; C_t is direct capital and operating cost; D_t is dependency and switching cost; R_t is the economic value of rights transferred or leaked; G_t is expected geopolitical, jurisdictional, or policy exposure; L_t is locally retained capability and option value; r is the appropriate discount rate; and ΔNRE is incremental engineering, integration, qualification, and transition cost.

The model deliberately does not assign universal weights. As in Analysis 001, a mystical score would create false precision. The objective is to expose the variables that must be estimated and attacked.

Exhibit 1. The Infrastructure Commitment Ladder

Structure	Relative dependency	Relative reversibility	Principal exposure
Portable service / commodity infrastructure	Low	High	Price, service continuity
Long-term cloud / managed platform	Moderate	Moderate	Egress, proprietary services, jurisdiction
Strategic finance + tied procurement	High	Low-moderate	Supplier and operating dependency
Integrated finance-build-operate stack	Very high	Low	Large dependency radius if assumptions fail

3. Three Different Sovereignties

Data sovereignty, compute sovereignty, and economic sovereignty are related but not interchangeable.

Data sovereignty concerns where data resides, which laws govern it, and who may lawfully access or process it. Compute sovereignty concerns who controls the resources, firmware, accelerators, scheduling, keys, update authority, and operational continuity required to transform that data. Economic sovereignty concerns where durable value accumulates: intellectual property, operating margin, engineering capability, derivative models, procurement leverage, supplier knowledge, and the ability to switch or renegotiate.

A system can satisfy a residency rule while remaining deeply dependent on foreign compute, foreign management planes, proprietary formats, externally controlled keys, or non-portable accelerators. Conversely, domestic ownership of server racks does not create meaningful sovereignty if firmware, toolchains, specialist operations, or critical components remain impossible to substitute.

Data sovereignty ≠ Compute sovereignty ≠ Economic sovereignty

The architectural objective is not autarky. It is intentional allocation of control and option value.

4. The Capital Stack Changes the Feasible Architecture

The World Bank's 2026 AI-infrastructure work emphasizes coordinated development of supply, demand, and rules, including public-sector reform, private financing, guarantees, cloud and data-center investment, connectivity, cybersecurity, interoperability, and skills.[3] That matters because infrastructure finance is not simply money arriving before architecture. It changes which architecture can be financed.

China is an especially important comparator because its Africa strategy can combine credit, assistance, corporate investment, infrastructure delivery, equipment, operations, training, and broader bilateral relationships. The 2025–2027 FOCAC Action Plan provides for RMB 360 billion of financial support across credit lines, assistance, and Chinese-company investment and explicitly includes digital infrastructure and digital-economy cooperation.[5]

A small clinical-compute company cannot compete with that balance sheet. Nor should the analysis pretend otherwise. Its potential advantage lies elsewhere: workload-specific architecture, clinical specialization, portability, transparent rights allocation, local capability formation, and preservation of future supplier choice.

The same test must be applied to Western hyperscalers, development-finance institutions, U.S. infrastructure vendors, regional providers, and domestic public capital. Nationality is not the risk variable. Dependency structure is.

Exhibit 2. Attack Every Arrow

Capital source → Rights → Vendor / infrastructure → Compute control → Clinical architecture → Operating dependency → Economic capture

Technical and investment diligence should interrogate every transition. A favorable financing rate does not automatically produce favorable economics if the financing transfers the learning curve, restricts procurement, enlarges switching cost, or makes the clinical system operationally inseparable from one provider.

5. Clinical Compute Makes the Problem Harder

Clinical systems cannot be evaluated as ordinary enterprise workloads. A hospital cannot safely assume perfect wide-area connectivity, unlimited cloud availability, or unconstrained data movement. High-consequence workloads impose continuity, provenance, latency, privacy, audit, and failure-mode requirements that feed backward into the infrastructure topology.

A generalized architecture therefore begins at the mission rather than the data center.

Clinical mission → Signal → Bedside / edge → Transport → Regional compute → Accelerator → Clinical output

The bedside tier may need to acquire and timestamp signals, preserve quality metadata, perform bounded preprocessing, execute minimum local inference, buffer during network loss, enforce device and software admission, and retain sufficient evidence to reconstruct which state produced an output. The regional tier may provide longitudinal context, higher-cost inference, fleet operations, model distribution, audit retention, and shared accelerator utilization.

Satellite and other wide-area connectivity can materially extend reach, but connectivity should be modeled as a stochastic resource rather than an always-on backplane. Maximum tolerable outage, queue depth, duplicate handling, replay protection, synchronization after reconnection, and degraded clinical behavior are architectural quantities. Connectivity economics therefore affect memory, state machines, inference partitioning, and clinical claims—not merely the networking budget.

6. Trust Determines What May Enter the Compute Path

Analysis 001 treated trust as an admissibility problem: evidence required before a signal, component, computational resource, or result may enter a consequential decision path.[7] That framing becomes more important when infrastructure is distributed across institutions and jurisdictions.

Kenya's July 2026 Guidance Note on Emerging Technologies identifies limited visibility and control, foreign-government access, sub-processor chains, data portability and vendor lock-in, retention, and cross-border processing as cloud-computing concerns.[6] These are not merely legal abstractions. They map directly to architecture: key custody, logging, export interfaces, data format, service decomposition, subprocessor boundaries, and the ability to revoke or migrate a workload.

A clinically consequential compute system should be able to answer which device produced an input; whether the device and software state were authorized; whether the input is fresh; which compute resource and model produced the inference; whether that resource remains admitted; and who possesses the authority to revoke, quarantine, update, or restore it.

Digital sovereignty without technical admission and exit controls can become sovereignty by contract rather than sovereignty by architecture.

7. What Should Become Local, Regional, Programmable, or Fixed?

A sovereignty argument is not a silicon argument. The substrate decision still has to clear the evidentiary gates in Analysis 001.[7]

Software should retain functions whose algorithms, interfaces, or operating assumptions remain volatile. Programmable hardware becomes attractive when deterministic latency, custom I/O, hardware isolation, parallelism, or measured efficiency matter while the implementation still needs to evolve. Fixed silicon becomes defensible only when the computation is persistent, the technical advantage is measured, lifetime economics clear the required hurdle, and the manufacturing and deployment path is admissible.

$$NPV_{fixed} - NPV_{flexible} = \Sigma_t \left[(Q_t \times \Delta CM_t - \Delta OPEX_t)/(1+r)^t\right] - \Delta NRE - E[L]$$

The Africa infrastructure problem adds another term to E[L]: the cost of architectural dependence when the compute platform, capital provider, or policy environment changes. The silicon decision therefore cannot be isolated from the infrastructure decision that will host the silicon.

8. NRE in Sovereign Clinical Compute: What Are We Actually Paying For?

NRE - non-recurring engineering - is the one-time engineering, integration, verification, qualification, tooling, and deployment work required to make a particular architecture exist. In this scenario it is not synonymous with ASIC mask cost, and it is not one undifferentiated line called "engineering." The infrastructure program can incur NRE long before fixed silicon is justified.

For a distributed clinical-compute program, the NRE ledger should be decomposed into work packages whose beneficiaries and reuse differ materially. The central allocation question is: which party is causing the work, which party will own the resulting asset or learning, and how much of that work can be reused after this deployment?

8.1 Six NRE buckets

- A. Clinical / signal-path NRE. sensor and acquisition interfaces; timestamping and synchronization; waveform ingestion; preprocessing and feature pipelines; bedside failure behavior; local buffering; clinical workflow integration; and evidence required to validate that the system behaves correctly under degraded connectivity.

- B. Platform / accelerator NRE. FPGA/SoC architecture; RTL; memory and DMA architecture; host interfaces; device drivers; firmware; model-to-hardware mapping; synthesis; place-and-route; timing closure; resource closure; bitstream generation; board bring-up; performance characterization; and reusable accelerator infrastructure.

- C. Trust / admission NRE. device enrollment; secure-boot and measured-state integration; key custody; workload and image authorization; freshness and anti-replay state; provenance binding; quarantine/revocation; audit evidence; recovery; and policy enforcement across bedside and regional compute.

- D. Partner / infrastructure integration NRE. interfaces to a particular hospital system, network, satellite/WAN service, data-center environment, cloud, server platform, accelerator card, orchestration layer, telemetry stack, identity system, or partner-owned API. This bucket is where a platform company can accidentally subsidize somebody else's deployment.

- E. Verification, qualification, regulatory, and operationalization NRE. hardware and software verification; fault injection; cybersecurity testing; clinical validation support; safety evidence; interoperability testing; environmental and EMC work where applicable; manufacturing test development; deployment procedures; observability; incident response; training; documentation; and acceptance testing.

- F. Silicon-transition NRE - only if the substrate gate is earned. microarchitecture freeze; ASIC RTL hardening; DFT; physical design; IP licensing; verification expansion; emulation/prototyping; package and board redesign; masks; characterization; bring-up; yield learning; production test; qualification; and re-certification or revalidation caused by the silicon change.

8.2 The NRE ledger

Each work package i should carry a cost ledger and a rights ledger. At minimum, record baseline NRE N_i^0; partner-conditioned NRE N_i^p; incremental causal NRE ΔN_i^p; funding by party F_i; ownership/control O_i; beneficiary set; lifecycle support obligation; rights transferred X_i; and the reusable asset or learning retained. Reuse and partner specificity may remain descriptive fields in a workbook, but they are not the economic allocation rule.

8.2.1 Cost, funding, ownership, and value are different quantities

For work package i, first establish engineering cost from observable categories rather than a subjective allocation score:

$$N_i = L_i + T_i + V_i + Q_i + E_i$$

where L_i is engineering labor, T_i is tools and licensed IP, V_i is verification and test, Q_i is qualification or certification work attributable to the package, and E_i is external engineering, design-house, laboratory, vendor, or specialist expense. All terms are currency quantities over the same price basis. This is the cost ledger.

Next distinguish four quantities that are often collapsed in negotiations:

$$N_i \neq F_i \neq O_i \neq A_i$$

N_i is the cost of creating the work package; F_i is the amount funded by a particular party; O_i describes ownership and control of the resulting artifact or IP; and A_i is the economic value of the asset created. Funding a work package does not, by itself, establish ownership of the resulting platform asset, and cost does not establish value.

8.2.2 Incremental causation

The cleanest starting point for partner allocation is counterfactual causation. Define N_i^0 as the NRE required for the platform without the partner-specific requirement and N_i^p as the NRE with that requirement. Then:

$$\Delta N_i^p = \max(0, N_i^p - N_i^0)$$

ΔN_i^p is the incremental engineering cost caused by the partner requirement. If the requirement disappeared, this is the engineering spend that would disappear with it. It is therefore the natural starting point for partner-funded NRE. It does not determine IP ownership or the value of rights transferred.

8.2.3 Reusable asset value

Some partner-caused work can still create a reusable platform asset. Estimate that reusable value separately rather than hiding it inside a reuse score:

$$A_i = \sum_t \left[E(\Delta FCF_{i,t}) + E(\Delta NRE_{avoided,i,t}) + E(\Delta KCF_{i,t})\right] / (1+r_i)^t + ROV_i$$

where $\Delta FCF_{i,t}$ is incremental free cash flow enabled by reusable work package i in period t; $\Delta NRE_{avoided,i,t}$ is future engineering cost avoided by reuse; $\Delta KCF_{i,t}$ is cash-equivalent capability or learning value only where it is appropriable and not already embedded in ΔFCF; r_i is the risk-adjusted discount rate; and ROV_i is residual option value for opportunities not already represented in the explicit cash-flow forecast. Option value may not be used as a catch-all residual.

8.2.4 Rights transfer is a separate transaction

A partner may fund NRE and separately request source access, RTL, architectural disclosure, exclusivity, derivative rights, field-of-use rights, training methods, optimization knowledge, or rights to improvements. Let X_i denote the risk-adjusted present value of those rights to the platform owner:

$$X_i = X_{exclusivity,i} + X_{derivative,i} + X_{disclosure,i} + X_{internalization,i} + X_{field,i}$$

Each component is valued from the cash flows or options surrendered. For example:

$$X_{internalization,i} = P(\text{internalization} \mid \text{deal}) \times PV(\text{cash flows lost if the partner substitutes internally})$$

$$X\_exclusivity,i = PV(\text{cash flows from credible opportunities contractually foreclosed})$$

A right is therefore not priced by intuition or by equating it to engineering cost. It is priced from the economic opportunities it removes or reallocates, with scenario probabilities made explicit.

X_i is not NRE. Treating it as though it were included in funded engineering can transfer an asset whose expected value materially exceeds the engineering reimbursement.

8.2.5 Platform value of a funded-NRE arrangement

For a platform owner evaluating a partner-funded work package, a simplified incremental value expression is:

$$\Delta V\_platform,i = F\_i + A\_retained,i - N\_unfunded,i - X\_i - C\_support,i - D\_i$$

Proceed only when:

$$\Delta V\_platform,i >= H\_i$$

where H_i is the organization's required value hurdle for accepting the additional commitment. H_i may be expressed as a minimum NPV, minimum return multiple, or another governance threshold, but it must be specified before negotiation rather than invented after the result is known.

where F_i is partner funding received; A_i^retained is the risk-adjusted value of the reusable asset retained by the platform owner; N_i^unfunded is engineering cost still borne by the platform owner; X_i is rights value transferred; C_i^support is incremental lifecycle support burden; and D_i is incremental dependency or switching cost created by the arrangement. A funded-NRE agreement is economically attractive only if this incremental value clears the organization's required hurdle.

8.2.6 Allocation rule

The defensible allocation rule is sequential rather than score-based:

1. Establish the baseline platform NRE that would exist without the partner.
2. Calculate the incremental NRE caused by the partner requirement.
3. Identify any reusable asset created by that incremental work and estimate its retained value without double counting.
4. Price separately any rights, exclusivity, disclosure, derivative learning, or option value transferred.
5. Allocate lifecycle support, qualification, and exit obligations to the party causing or benefiting from them.
6. Release engineering capital only when the resulting retained value clears the applicable hurdle.

This formulation replaces the earlier reuse-factor heuristic as the primary economic model. Reuse and partner specificity remain useful descriptive fields in the NRE ledger, but they no longer determine funding by arbitrary multiplication.

8.2.7 Units, uncertainty, and double-counting discipline

All NRE, rights, support, dependency, and asset-value terms are expressed in currency on a common valuation date. Future values are discounted using a rate consistent with the risk of the cash flow being modeled. Technical probabilities such as deployment success, reuse probability, or qualification success enter through scenario probabilities or expected values rather than being added directly to currency terms. Maintain a term map showing where each economic effect enters the model. If switching cost is included in D_i, the same expected redesign cost must not also appear in asset value, project loss, or exit cost. If future contribution margin is included in A_i, it must not be counted again in the parent project NPV.

8.3 NRE is not the same as capex or opex

The model should keep three ledgers separate. NRE is the one-time engineering required to create or adapt the system. Capex purchases durable assets such as servers, accelerator cards, bedside hardware, networking equipment, racks, power systems, or data-center capacity. Opex pays for recurring operation: bandwidth, cloud consumption, colocation, power, support, monitoring, maintenance, personnel, licenses, and hardware replacement. Blurring these categories can make a financing offer look cheaper by moving cost from one ledger to another.

8.4 A worked hypothetical allocation

Consider a hypothetical distributed clinical-compute platform. Before any specific infrastructure partner, the company expects $1.20 million of reusable platform NRE. A proposed deployment introduces four incremental work packages: a proprietary infrastructure adapter, production admission/security integration, deployment-specific verification, and qualification/integration work. Those incremental costs are caused by the relationship and therefore form the initial partner-NRE claim.

Work package	Baseline NRE	With partner	Incremental causal NRE	Primary beneficiary
Reusable platform architecture	$1,200,000	$1,200,000	$0	Platform
Proprietary infrastructure adapter	$0	$300,000	$300,000	Partner relationship
Production admission / security integration	$0	$180,000	$180,000	Shared; reusable elements possible
Deployment-specific verification	$0	$120,000	$120,000	Deployment / partner
Qualification and integration	$0	$200,000	$200,000	Deployment / partner
TOTAL	$1,200,000	$2,000,000	$800,000	-

Assume the partner funds the full $800,000 incremental NRE. That does not settle the economics. Suppose the work also creates $450,000 of risk-adjusted reusable asset value retained by the platform owner, but the partner requests rights with a risk-adjusted value of $1.80 million; the expected lifecycle support burden is $125,000; and expected dependency loss is $350,000. The platform owner has no unfunded incremental engineering in this example.

$$\Delta V\_platform = 800{,}000 + 450{,}000 - 0 - 1{,}800{,}000 - 125{,}000 - 350{,}000$$

$$\Delta V\_platform = -1{,}025{,}000$$

The phrase “the partner is paying all the NRE” is therefore economically incomplete. Narrow the requested rights, preserve portability, or change the support/dependency structure and the same $800,000 of funded engineering can become attractive. The model makes the negotiation variable explicit rather than hiding it inside the engineering reimbursement.

8.5 NRE can be consideration in a strategic transaction

Funded NRE is economically meaningful consideration. If a strategic partner funds engineering that accelerates a reusable platform, that funding can reduce the platform company's capital burden without transferring ownership. But funded NRE should not silently purchase source code, RTL, derivative rights, exclusivity, unrestricted implementation knowledge, or a right to internalize the architecture. Those rights require separate valuation. Conversely, if a partner is paying for genuinely partner-specific work, the payment is not a favor or strategic investment; it is the economic cost of the requested integration.

8.6 NRE release gates

NRE should be released in stages. Interface-definition NRE should precede implementation NRE. FPGA evidence should precede ASIC-transition NRE. Verification evidence should precede production qualification. Partner-specific NRE should not begin until interface ownership, deliverables, acceptance criteria, IP rights, and change-control responsibility are explicit. This converts NRE from an engineering burn rate into a sequence of capital commitments tied to evidence.

9. Risk-Adjusted Economic Capture

The relevant output is not simply project NPV. It is the portion of lifetime value retained by the actors whose capital, policy, clinical access, infrastructure, and engineering made the system possible.

Economic capture includes cash margin but also option value: reusable IP, operational learning, data-improvement rights where lawful, engineering capability, supplier knowledge, installed interfaces, switching leverage, and the ability to build the next system at lower marginal cost.

$$\text{Retained Value}_j = V_j - PV(\text{value transferred outside the chosen beneficiary set})$$

The beneficiary set must be named. A national policymaker may define retained value around domestic capability and control; a platform company may define it around company-owned IP, margin, and option value; a hospital may define it around clinical continuity and lifecycle cost. “Economic capture” is therefore perspective-dependent, but the cash-flow mechanics remain explicit.

A financing structure can therefore produce positive project NPV and still be strategically unattractive if most of the compounding assets leave with the provider. Conversely, paying somewhat more for infrastructure can be rational if the premium buys portability, local capability, key control, reusable engineering, and credible exit rights.

9.1 Capital-stack comparison: what must be estimated

Structure	Financing cost	Dependency drivers	Rights/control	Capability retained	Key sensitivity
Integrated strategic finance/build/operate	Potentially low/competitive	Tied procurement, operating stack, switching	Deal-specific	Deal-specific	Exit cost and control rights
Hyperscaler / managed cloud	Low initial capex	Egress, proprietary services, jurisdiction	Service terms	Moderate	Usage growth and portability
Blended / DFI-backed infrastructure	Potentially lower WACC	Governance and execution complexity	Usually negotiated	Potentially high	Guarantees, local operating capability
Modular sovereign/private stack	Potentially higher upfront cost	Integration burden, multi-vendor operations	Can preserve control	Potentially high	Utilization, skills, coordination

The table is not a ranking. It identifies which terms must be populated for each structure. The preferred structure is the one with the highest risk-adjusted value for the relevant beneficiary set after rights, dependency, capability, and exit are valued consistently.

9.2 Clinical benefit: monetize only what can be defended

Clinical benefit should not be converted into dollars merely to make unlike outcomes fit one spreadsheet. The model distinguishes three cases. First, where a clinical effect produces a measurable cash consequence - for example avoided transfer, avoided duplicate testing, reduced infrastructure use, reduced length of stay when causally supportable, or another payer/provider cash effect - that incremental cash flow may enter CF_clinical. Second, where the effect is clinically important but does not have a defensible cash-flow proxy, it remains a mission outcome or constraint and is reported alongside NPV rather than monetized. Third, where monetization would require a value-of-life, value-of-statistical-life, QALY, or health-utility framework, the paper does not invent one; the applicable health-economic or public-policy framework must be supplied by the decision maker.

A higher clinical benefit does not automatically rescue an economically or architecturally inadmissible structure. Likewise, a structure with the highest financial NPV may still be rejected when it fails a non-negotiable clinical continuity, safety, privacy, or sovereignty constraint.

9.3 Discount rates: match the rate to the risk being discounted

The model does not require one universal discount rate. Contracted partner reimbursements, speculative product margin, government-backed infrastructure cash flows, and residual option value do not carry the same risk. Where separate cash-flow classes have materially different risk, either model them in separate scenarios or apply risk-appropriate discount rates. Do not simultaneously reduce a cash flow by a probability of success and then add a large risk premium for the same uncertainty without documenting why both adjustments are required.

For decision-room use, every rate must therefore answer three questions: what cash flow is being discounted; what risks the rate is intended to compensate; and which risks are modeled explicitly elsewhere. If those questions cannot be answered, the rate is not an input - it is a hidden assumption.

9.4 Geopolitical and policy risk: scenarios, not nationality penalties

Geopolitical exposure should enter the model through identifiable mechanisms: export restrictions, sanctions, forced supplier substitution, cross-border data restrictions, currency convertibility, tied procurement, change in tax or import treatment, loss of support, local-content rules, licensing delay, or other events that alter cash flow or architecture. Nationality by itself is not a cost term.

For structure j and scenario g, the economic treatment is:

$$G_j = \sum_g P(g|j) \times PV(C_{transition,g} + C_{delay,g} + C_{substitution,g} + \Delta CF_g)$$

where the cost terms describe transition, delay, and substitution consequences and ΔCF_g captures the incremental cash-flow effect of the policy event. If the same redesign or switching cost is already included in dependency loss D_j, it must not also be charged to G_j. In practice, the analyst may place a scenario in D_j or G_j according to its causal origin, but not both.

9.5 Rights and internalization: use scenario valuation, not false precision

Rights-transfer value is most uncertain when the transferred right changes future bargaining power rather than an already-contracted cash flow. The model therefore prefers scenarios over point estimates. For example, internalization risk should be expressed as a set of explicit states - no internalization, partial substitution, or full substitution - each with a probability range and a defined cash-flow consequence. The purpose is not to claim that management knows a 17.3 percent probability of internalization. The purpose is to expose how much the decision changes if the probability is 5, 20, or 50 percent.

If the recommendation reverses under a modest change in a poorly observable probability, that uncertainty is itself a decision result: the transaction requires stronger contractual protection, additional guaranteed economics, or delayed commitment.

9.6 Capital-room reporting: show ranges, constraints, and reversal points

A decision-grade output should therefore report at least: the base case; downside and upside scenarios; non-negotiable mission or sovereignty constraints; the variables with the greatest influence on the result; and the reversal point at which the preferred structure changes. A single NPV is not the conclusion. The conclusion is the range of conditions under which a structure remains superior to its alternatives.

10. What Must Remain True?

Every infrastructure investment embeds a forecast just as every custom-silicon investment does. It forecasts clinical demand, connectivity, power, workload growth, policy stability, financing availability, supplier behavior, operating competence, and the durability of the chosen interfaces.

Diligence should therefore ask what must remain true for the capital committed to this architecture to earn its return.

- Which clinical outcome or economic constraint actually requires this compute investment?

- What minimum capability must survive loss of the wide-area network?

- Which party controls keys, firmware approval, workload admission, model approval, logs, and recovery?

- What rights are transferred with the financing, and what are those rights economically worth?

- What engineering knowledge remains locally after the relationship ends?

- How much of the system must be replaced if the infrastructure provider changes?

- What recurring workload evidence justifies FPGA acceleration or fixed silicon?

- Which NRE is reusable platform investment and which is deployment-specific subsidy?

- What capital or policy change would invalidate the architecture?

- What evidence would cause management not to accept the financing or not to build the silicon?

11. What Would Cause Us Not to Take the Capital?

This may be the most revealing question in the paper.

Do not accept the structure if tied procurement eliminates credible supplier substitution and the financing advantage does not compensate for the resulting dependency. Do not accept it if the provider controls operational keys or update authority in a way inconsistent with the mission's sovereignty or continuity requirements. Do not transfer broad derivative or implementation rights merely to reduce near-term NRE unless their risk-adjusted value has been priced. Do not centralize a clinical function whose minimum safe behavior must survive network loss. Do not buy accelerator capacity because 'AI requires GPUs' when measured workload economics do not justify it. Do not tape out merely because capital is available.

The exact thresholds are project-specific. The discipline is requiring the kill conditions to exist before the dependency is created.

12. Policy Is Part of the Feasible Design Space

Policy does not become an RTL module. It can nevertheless change which architectures are economically or legally feasible. Data-transfer rules affect storage and processing location. Cybersecurity requirements affect identity, key control, logging, and update architecture. Public guarantees and blended finance can lower the cost of shared compute and power infrastructure. Procurement rules can either increase lock-in or require portability. Local-capability requirements can convert an infrastructure purchase into engineering transfer rather than perpetual external dependence.

The strongest policy is therefore not necessarily the policy that mandates domestic ownership of every layer. An outcome-based approach can instead require measurable control: portability, key custody, documented exit, operational continuity, transparent subprocessor boundaries, local capability, and the ability to substitute providers without rebuilding the clinical system.[4]

This is where policy and architecture meet. Policy defines the outcomes that must remain possible. Architecture determines whether those outcomes are technically real.

13. Beyond Africa: Sovereignty as Infrastructure Investment

The African reference case makes the interaction among capital scarcity, infrastructure gaps, external financing, clinical urgency, and sovereignty unusually visible. The underlying architecture problem is not uniquely African. Jurisdictions with substantially greater capital formation can still decide that dependence on external compute, cloud, semiconductor, or software stacks creates an unacceptable strategic exposure.

Europe is a useful comparator because the policy response is explicit. In June 2026 the European Commission proposed a technological-sovereignty package spanning semiconductors, cloud, AI, and open source. The package includes a proposed Chips Act 2.0 and Cloud and AI Development Act, and frames the objective as strengthening European capacity, choice, autonomy, and resilience in core technologies.[9][10]

The infrastructure commitment is also material. In July 2026 the EU launched a call for up to seven AI Gigafactories intended to mobilize more than EUR 30 billion of investment. EuroHPC describes the facilities as integrated AI-compute environments combining AI-optimized supercomputers, advanced data centers, high-capacity storage, ultra-fast networks, secure cloud access, and specialized support services.[11][12] The Commission separately identifies advanced-chip sovereignty and supply security as important for AI and cloud infrastructure and for strategic sectors including health, defense, robotics, secure communications, and autonomous systems.[13]

13.1 Sovereignty has a capital cost

This comparison exposes an important point: sovereignty is not created by declaration. It is an engineered capability that requires capital, NRE, operating capability, migration effort, and in some cases a willingness to pay a recurring premium for optionality and resilience.

$$C\_sovereignty = NRE\_transition + CAPEX\_local + PV(OPEX\_premium + C\_migration + C\_capability)$$

C_sovereignty is the incremental cost of creating the desired sovereign capability relative to the best less-sovereign alternative. NRE_transition includes architecture, integration, porting, validation, security, and qualification work required to move or duplicate capability. CAPEX_local is incremental durable infrastructure. OPEX_premium is any recurring cost premium. C_migration is the expected cost of moving data, workloads, interfaces, and operations. C_capability is the cost of building and sustaining the people, processes, tools, and operating competence needed to exercise the control being purchased.

The existence of this cost does not make sovereignty irrational. It makes sovereignty an investment thesis that must identify the economic and strategic benefit it is buying.

$$SV\_sovereignty = PV(D\_avoided + R\_resilience + EC\_retained + O\_portability) - C\_sovereignty$$

Here D_avoided is expected dependency loss avoided; R_resilience is the cash-equivalent value of continuity or resilience where it can be defensibly monetized; EC_retained is additional economic capture retained domestically or by the intended platform owner; and O_portability is residual option value from credible supplier substitution or future strategic choice. As elsewhere in this paper, terms that cannot be defensibly monetized should remain explicit constraints or scenario outcomes rather than being assigned arbitrary dollar values.

13.2 The same framework, different starting conditions

Africa and Europe therefore represent different starting conditions rather than different theories. In many African markets, insufficient domestic capital formation can increase dependency because infrastructure must be financed externally. In Europe, the policy problem can arise despite substantial capital: existing dependence on non-European cloud, semiconductor, software, and compute providers can make strategic autonomy itself an investment objective. The United States faces still another configuration, with stronger domestic platform capacity but material dependencies in foundry, advanced packaging, materials, and global supply chains.

The general architecture question is the same: what should be owned, what can be rented, which control surfaces must remain substitutable, and which dependency becomes strategically unacceptable?

13.3 Sovereignty by architecture rather than ownership alone

The comparative lesson is that sovereignty should be evaluated by capability rather than by the nationality printed on an invoice or the nominal owner of a server. An architecture can preserve meaningful sovereignty through local key custody, workload and device admission authority, documented interfaces, portable data and models, multi-vendor substitution, operational competence, bounded external rights, and credible exit procedures even when parts of the physical infrastructure are externally financed or operated.

Conversely, domestic ownership can provide little practical sovereignty when the critical software, firmware, accelerators, toolchains, management plane, or specialist operating knowledge cannot be substituted. The relevant unit is therefore control over consequential functions and the cost of exercising exit - not symbolic ownership.

13.4 Policy implication: specify measurable control outcomes

This suggests a policy discipline that generalizes beyond the African reference case. Governments and institutional buyers can specify measurable outcomes - portability, key custody, auditability, local operating capability, workload continuity, interface transparency, bounded data and derivative rights, and tested exit procedures - without requiring domestic ownership of every layer. Architecture then determines whether those outcomes are technically real, and the capital model determines what they cost.

The result is a broader interpretation of infrastructure policy: sovereignty is a portfolio of engineered options whose value should be compared with the capital required to preserve them.

14. Evidence Before Infrastructure Irreversibility

A decision framework for sovereign clinical compute should not produce a universal sovereignty score. It should produce gates.

Gate 1: Mission necessity. Identify the clinical or economic outcome and the minimum local capability required to achieve it.

Gate 2: Infrastructure feasibility. Establish power, connectivity, data movement, operational support, security, and failure-mode requirements.

Gate 3: Capital admissibility. Compare financing structures using direct cost, rights transferred, dependency radius, geopolitical/policy exposure, and locally retained capability.

Gate 4: Control and portability. Demonstrate key custody, workload and device admission, exportability, interface ownership, recovery authority, and a credible provider-exit path.

Gate 5: Substrate evidence. Apply the measured-advantage, persistence, risk-adjusted lifetime economics, and implementation-admissibility gates from Analysis 001 before increasing hardware irreversibility.[7]

If the system fails a gate, the answer is not automatically to abandon the program. It is to preserve optionality at that layer until the missing evidence exists.

Conclusion: What Deserves to Become Dependent?

Africa's AI infrastructure build-out will require substantial external capital, technology, and expertise. Treating that fact as either salvation or exploitation is analytically weak. The consequential question is what each financing and technology structure causes the recipient to own, control, learn, retain, and become unable to replace.

The lowest nominal cost of capital can become expensive if it transfers the learning curve. Domestic ownership can become symbolic if the keys, firmware, tools, or operational expertise remain externally controlled. Data residency can coexist with compute dependency. Custom silicon can destroy capital if the workload has not earned irreversibility. And a clinically elegant cloud architecture can become unsafe if the network is treated as more reliable than the deployment permits.

The objective is not to resist foreign capital or specialization. Both can create extraordinary value.

The discipline is to match the dependency created by the financing to the evidence that the dependency is worth accepting.

The question is therefore not merely who will pay for the compute.

It is what deserves to become dependent.

Appendix A. Mathematical Derivation and Model Governance

A.1 NRE cost identity

$$N_i = L_i + T_i + V_i + Q_i + E_i$$

Currency. Inputs: loaded labor plan, EDA/IP/tool quotes, verification/test plan, qualification plan, external vendor/design-house/lab quotes. Interpretation: observable one-time creation cost. No probabilities appear in this identity.

A.2 Incremental causal NRE

$$\Delta N_i^p = \max(0, N_i^p - N_i^0)$$

Currency. Inputs: baseline platform scope and partner-conditioned scope. Interpretation: engineering spend that disappears if the partner requirement disappears. Any cost savings caused by the partner are recorded separately rather than as negative NRE.

A.3 Reusable asset value

$$A_i = \text{sum}_t \left[E(\Delta FCF_{i,t})+E(\Delta NRE_{avoided,i,t})+E(\Delta KCF_{i,t})\right]/(1+r_i)^t + ROV_i$$

Currency present value. Inputs: risk-adjusted incremental free cash flow, future NRE avoided, appropriable capability cash-equivalent, discount rate, residual option scenarios. Governance: no future product may appear both in ΔFCF and ROV.

A.4 Rights-transfer value

$$X_i = \Sigma_k X_{i,k}$$

Currency present value. Inputs: exclusivity, derivative rights, disclosure, internalization, field-of-use or other rights. Each component is valued from cash flows/options surrendered, not from engineering cost.

A.5 Expected dependency loss

$$D_j = \Sigma_s P(s|j) \times PV(C_{switch,s}+C_{redesign,s}+C_{requal,s}+C_{downtime,s})$$

Currency expected present loss. Inputs: failure/migration scenarios, conditional probabilities, switching/redesign/requalification/downtime costs. Governance: overlapping scenarios require a scenario tree or conditional treatment.

A.6 Platform value of funded NRE

$$\Delta V_{platform,i} = F_i + A_{retained,i} - N_{unfunded,i} - X_i - C_{support,i} - D_i$$

Currency present value. Inputs: funded NRE, retained reusable asset value, unfunded engineering, rights transfer, lifecycle support, dependency loss. Decision: proceed only if $\Delta V \geq H_i$.

A.7 Structure-level value

$$V_j = \Sigma_t E[CF_{clinical}+CF_{commercial}+CF_{capability}-C_{infra}-C_{dependency}-C_{rights}]/(1+r_j)^t + ROV_j$$

Currency present value. Inputs depend on beneficiary perspective. Governance: each economic effect appears once; residual option value includes only opportunities outside explicit forecasts.

A.8 Double-counting rule

Maintain a term map for every scenario. If an effect is already embedded in forecast free cash flow, it cannot also appear as capability value, dependency loss, rights loss, exit cost, or option value. If a switching/redesign cost appears in D_j, do not also subtract it in a separate exit-cost line. If partner funding reimburses an NRE item, the reimbursed amount reduces N_unfunded rather than creating a second benefit unless it is a true fee above cost.

A.9 Clinical-benefit treatment

Clinical outcomes enter the currency model only when a defensible incremental cash-flow or avoided-cost proxy exists and causal attribution is supportable. Otherwise they are reported as non-monetized mission outcomes or constraints. Health-utility or value-of-life monetization requires an externally selected health-economic framework and is not inferred by this architecture model.

A.10 Discount-rate governance

Rates must identify the cash-flow class and the risk compensated by the rate. Risks modeled explicitly through scenario probabilities should not be silently charged a second time through an undocumented premium. Different cash-flow classes may use different rates or separate probability-weighted scenarios when their risk profiles differ materially.

A.11 Geopolitical / policy scenario loss

$$G_j = \Sigma_g P(g|j) \times PV(C_{transition,g}+C_{delay,g}+C_{substitution,g}+\Delta CF_g)$$

Currency expected present effect. Inputs: defined policy/geopolitical event states, conditional probabilities or probability ranges, transition/delay/substitution costs and incremental cash-flow effects. Governance: nationality is never itself a penalty; overlapping switching/redesign losses cannot also appear in D_j.

A.12 Reversal-point analysis

For uncertain input θ, solve or search for θ* such that V_A(θ*) = V_B(θ*). θ* is the decision reversal point. Reporting θ* is often more informative than arguing over a single forecast because it states exactly how wrong an assumption must be before the preferred structure changes.

A.13 Sensitivity and model governance

The most decision-sensitive inputs should be varied explicitly: deployment probability, utilization, volume, discount rate, partner funding, switching probability, redesign/requalification cost, support burden, rights scope, and residual option value. The public paper should show scenario ranges rather than imply point-estimate certainty. A model output is evidence for a decision, not a substitute for governance.

References

1. Martin Schindler, Andrew J. Tiffin, Andrea Richter Hume, et al., “Unlocking the Potential: AI in Sub-Saharan Africa,” IMF Departmental Papers 2026, No. 013, July 21, 2026. DOI: 10.5089/9798229047821.087.A001. https://www.elibrary.imf.org/view/journals/087/2026/013/article-A001-en.xml

2. Ambrose Wong, Jeremy Ng, Taylor Reynolds, David Satola, and Aubra Anthony, “Moving toward an outcome-based approach for digital sovereignty,” World Bank, June 23, 2026. https://blogs.worldbank.org/en/digital-development/moving-toward-an-outcome-based-approach-for-digital-sovereignty-

3. World Bank Group, “Building Data Infrastructure for AI Readiness,” May 6, 2026. https://www.worldbank.org/en/results/2026/05/06/data-infrastructure-for-ai

4. World Bank Group, “Trust by Design: Securing Digital Development,” June 9, 2026. https://www.worldbank.org/en/results/2026/06/09/trust-by-design-securing-digital-development

5. Forum on China-Africa Cooperation, “Forum on China-Africa Cooperation Beijing Action Plan (2025–2027),” September 6, 2024. https://2024focacsummit.mfa.gov.cn/eng/hyqk_1/202409/t20240906_11486213.htm

6. Office of the Data Protection Commissioner, Kenya, “Guidance Note on Emerging Technologies,” July 2026. https://www.odpc.go.ke/wp-content/uploads/2026/07/Emerging-Tech-Guidance-Note.-July-2026.pdf

7. 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.

8. 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. Corrected version DOI: 10.5281/zenodo.22019132.

9. European Commission, “Strengthening Europe’s Tech Sovereignty,” 2026. https://digital-strategy.ec.europa.eu/en/policies/eu-tech-sovereignty

10. European Commission, “Commission proposes tech sovereignty package to strengthen Europe’s digital autonomy and resilience,” June 3, 2026. https://digital-strategy.ec.europa.eu/en/news/commission-proposes-tech-sovereignty-package-strengthen-europes-digital-autonomy-and-resilience

11. European Commission, “EU launches AI Gigafactories call to boost Europe’s computing capacity and unlock more than €30 billion in investment,” July 30, 2026. https://digital-strategy.ec.europa.eu/en/news/eu-launches-ai-gigafactories-call-boost-europes-computing-capacity-and-unlock-more-eu30-billion

12. European High Performance Computing Joint Undertaking (EuroHPC JU), “The EuroHPC Joint Undertaking launches the AI Gigafactories Call,” July 30, 2026. https://www.eurohpc-ju.europa.eu/eurohpc-joint-undertaking-launches-ai-gigafactories-call-2026-07-30_en

13. European Commission, “Advanced chips,” 2026. https://commission.europa.eu/topics/competitiveness/competitiveness-coordination-tool-projects/advanced-chips_en

Author Note

Dana Xiadani is Founder & Chief Architect of Living Cipher, where she works at the boundary of computation, architecture, trust, physical realization, and capital allocation. This paper presents a public analytical framework for infrastructure and clinical compute. It does not describe or disclose any confidential client, partner, hospital, financing, deployment, reserved capacity, or NDA-protected architecture. Living Cipher proprietary scoring methods, implementation mechanisms, and protected architecture are not disclosed.

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

 The analysis uses standard discounted-cash-flow, incremental-cost, expected-loss, and scenario methods to expose the capital consequences of architecture decisions. It is not presented as a macroeconomic forecast, health-economic valuation standard, or substitute for ministry, DFI, clinical, legal, tax, or sovereign-finance expertise.