IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 13, Issue 9, September 2026

DOI: 10.17148/IARJSET.2026.13918

The Anchor-and-Expand Principle: Timing Strategic Decoupling in Technology Ecosystems

Vivek Gujar

Chief Strategy Officer, Indo AI Technologies Pvt. Ltd., Pune, Maharashtra, India

Abstract: Technology ecosystems often reward whichever firm first controls a tightly held physical or infrastructural asset — the "anchor" — because that control reduces uncertainty, sets de facto standards, and buys market access for a higher-margin software, service, or workflow layer — the "expansion." Existing strategy research explains much of why this pattern works, drawing on transaction cost economics, complementary-assets theory, the resource-based view, platform strategy, and dominant-design theory. What it says less about is timing: exactly when an anchor's protective value starts to fade, and when a firm should decouple from it.

This paper develops the Anchor-and-Expand (A-E) Principle and introduces Anchor Half-Life (t½) as a way of describing how long an anchor provides real strategic cover before imitation, standardization, or commoditization erodes it. The framework rests on four linked elements — Anchor Asset (A), Expansion Layer (E), Diffusion Threshold (D) and Anchor Half-Life (t½) — and three propositions concerning anchor value under architectural uncertainty, the "coupling trap" created by staying attached to an anchor too long and the compression of anchor advantage as technology cycles accelerate. Six historical cases — mainframes, personal computing, mobile platforms, cloud infrastructure, software-defined computing and edge AI — illustrate the framework and its limits.

The analysis suggests anchor ownership is most valuable during periods of high uncertainty, but that durable value increasingly comes from making the expansion layer portable across many anchors before diffusion and commoditization take hold. The paper is conceptual rather than empirical: its t½ estimates are directional and illustrative, and the IndoAI case is offered as an application of the framework rather than as independent empirical validation. It closes with a research agenda for measuring t½, gauging diffusion thresholds, and testing the framework across industries.

Keywords: Appization, AI Advisor, AI agents, CCTV Implementation, Edgebox, Edge Computing, Indoai

1. INTRODUCTION & CONCEPTUAL FOUNDATIONS

In emerging technology regimes where the architecture itself is still unsettled, participants face high transaction costs, undefined interfaces, and coordination failure. Under these conditions, whoever controls a minimal but essential piece of infrastructure — a hardware baseline, a reference design, a core protocol — gets to set the boundary conditions for everyone else building on top of it.

This is a familiar mechanic from consumer markets: razor-and-blades pricing, or printer-and-ink economics, where the base unit is priced low to lock in a long tail of higher-margin consumables [1]. In complex technology ecosystems the same logic plays out, but with much higher stakes:

- **The Anchor** is the baseline asset — it sets performance benchmarks, enforces interoperability standards, and reduces buyer risk while the category is still new.
- **The Expansion Layer** is the software, analytics, or service architecture that rides on the anchor's early market access to build higher-margin, recurring revenue.

Firms that anchor a market this way shape where value gets created across the whole ecosystem — and, just as often, who ends up dependent on whom [2]. Our premise is that anchor-first firms carry a structural risk: over-reliance on physical control. As prototyping tools, open standards, and modular fabrication mature, the window of anchor-derived advantage — the Anchor Half-Life — keeps getting shorter with each successive technology wave.

This paper is conceptual. It develops the Anchor-and-Expand (A-E) principle as a way of thinking about when a firm should establish control through a technological anchor, and when it should strategically decouple its expansion layer

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 9, September 2026

DOI: 10.17148/IARJSET.2026.13918

instead. The historical cases here are used for illustration rather than statistical hypothesis testing, and the half-life (t½) estimates throughout are directional interpretations of the historical record rather than empirically estimated parameters.

2. THEORETICAL FRAMEWORK & SYNTHESIS

The Anchor-and-Expand principle draws on five established strands of strategy research:

- **Transaction Cost Economics (TCE):** Coase [3] first asked why firms internalize production instead of relying on markets. Williamson [4, 5] answered that firms own assets when transaction uncertainty and asset specificity are high, which guards against opportunistic exploitation. Klein, Crawford, and Alchian [6] focused specifically on appropriable quasi-rents and the hold-up risk that comes with specialized assets — a concern later echoed in broader discussions of value creation and stakeholder governance [7].
- **Profiting from Innovation (PFI):** Teece [8–10] showed that core innovations are easy to imitate, so capturing value depends on controlling specialized complementary assets — manufacturing capacity, distribution networks, or proprietary hardware platforms.
- **Resource-Based View (RBV):** Barney [11], Peteraf [12], and Rumelt [13] framed competitive advantage around resources that are valuable, rare, hard to imitate, and non-substitutable. Rumelt [13] in particular pointed to "isolating mechanisms" — accumulated operational know-how, for instance — that keep fast followers from copying an incumbent's edge.
- **Platform Dynamics & Ecosystems:** Rochet and Tirole [14] laid out two-sided market dynamics, where a platform balances non-neutral pricing across distinct user groups. Eisenmann, Parker and Van Alstyne [15] described "platform envelopment" — entering an adjacent market by bundling capabilities with an incumbent's user base. Gawer and Cusumano [16], along with Iansiti and Levien [17], described "platform leadership" and "keystone strategies": core firms stay dominant by controlling a few central ecosystem nodes rather than owning every peripheral piece, building on Moore's [18] earlier work on business ecosystems. In a related B2B context, Gujar [19] argues that sustainable value in hardware-heavy marketplaces comes from moving away from low-margin transactions toward a software-enabled ecosystem that opens access to independent developers and captures data network effects — an argument developed further, alongside the "10Ps Framework" and "DUAL Architecture," in a companion review of platform growth strategy [20].

The Expansion Layer construct in this paper also builds on that earlier formalization of platform-level value capture in camera-based AI marketplaces [19], which separates a certified, outcome-oriented product layer from the underlying device layer — a distinction that maps closely onto the Anchor/Expansion split proposed here, developed independently and ahead of this paper's formalization of t½. A related extension applies the same DUAL Architecture and network-effects logic to the software layer specifically, modeling agentic computer vision as a shift from passive sensing to autonomous reasoning [21], and offers a more granular account of what "expansion" looks like once an anchor is already established.

- **Dynamic Capabilities & Dominant Designs:** Utterback and Abernathy [22] modeled the shift from fluid product innovation to process optimization once an industry settles on a "dominant design." Henderson and Clark [23] showed how architectural innovation can disrupt incumbents, and Teece, Pisano, and Shuen [24] described dynamic capabilities as a firm's ability to reconfigure its internal and external competencies.

Taken together, this literature explains why controlling an anchor is valuable — but says comparatively little about how long that control stays viable as technology cycles keep accelerating.

3. THE ANCHOR-AND-EXPAND (A-E) FRAMEWORK

The framework is meant as a conceptual sequencing model, not a predictive law. The central question isn't whether a firm should control a technological anchor at all — it's when that control stops being the most effective basis for value capture, and when resources should shift toward a portable expansion layer.

The A-E framework tracks the life cycle of architectural control across four elements:

- **Anchor Asset (A):** the minimal physical or structural unit that sets performance specifications for everything built on top of it.
- **Expansion Layer (E):** the software or service layer designed to run across many different hardware environments.
- **Diffusion Threshold (D):** the point at which interface standards solidify, open-source alternatives multiply, and hardware becomes a commodity.

DOI: 10.17148/IARJSET.2026.13918

- **Anchor Half-Life ($t½$):** how long it takes for an anchor's proprietary advantage to decay by half, through imitation, reverse engineering, or standards convergence.

Core Research Propositions

- **P1 — Architectural Uncertainty & Anchor Value:** under high architectural ambiguity, owning the anchor generates the greatest strategic rent, because it minimizes transaction costs and buyer risk.
- **P2 — Time-Boxed Advantage & The Coupling Trap:** value creation peaks just before the Diffusion Threshold (D). Continued capital spending on anchor hardware after D produces diminishing returns and becomes an infrastructure burden.
- **P3 — Paradigm Acceleration (Half-Life Compression):** advances in electronic design automation, open standards, and rapid prototyping shorten t½ with each successive technology wave. Long-run enterprise value tracks with how early a firm decouples its software.

3.1 A Simple Way to Read the Formula

Enterprise value in this framework is really just two separate "value pots" added together.

1. Anchor value starts high and shrinks by half every t½ years — the same shape as radioactive decay:

$$VA(t) = VA,0 \times (½)^{(t / t½)}$$

2. Expansion value starts near zero, grows slowly at first while it leans on the anchor's credibility, then accelerates and eventually levels off — the classic S-curve seen in technology adoption:

$$VE(t) = KE / (1 + e^{(-k(t - Xc))})$$

Total enterprise value at any time t is simply: **V(t) = VA(t) + VE(t)**

What the variables mean

- **VA,0 (Initial Anchor Value):** the early market power gained from controlling the physical hardware or reference design.
- **t½ (Anchor Half-Life):** how many years it takes competitors to copy or standardize the hardware, cutting its value in half.
- **KE (Expansion Ceiling):** the maximum value the software/service layer can reach once fully scaled.
- **k (Growth Rate):** how quickly the expansion layer picks up speed once it starts growing.
- **Xc (Expansion Crossover):** the point where software/service revenue overtakes hardware revenue — the moment E(t) = A(t).
- **D (Diffusion Threshold):** the point where hardware turns into a cheap, low-margin commodity.

In short: hardware value fades on a predictable clock (t½), while software value compounds on an S-curve. The strategic question is simply whether the expansion layer is far enough along that curve by the time the anchor's clock runs out.

Applying the Formula Across Eras

1. The Mainframe Era: IBM System/360 (Long Half-Life)

In 1964, building a computer from scratch was still extraordinarily hard. IBM built both the physical mainframe (A) and the operating system (E).

- **The math:** Anchor Half-Life (t½) ≈ 15–20 years. Competitors couldn't easily clone mainframes — chip design and manufacturing required massive capital.
- **The result:** because t½ was long, VA(t) decayed slowly. IBM earned high hardware margins for decades before it had to lean on software and services alone.

2. The Personal Computer Era: IBM PC (Accelerated Half-Life)

In 1981, IBM built the PC from off-the-shelf parts, keeping only the BIOS chip proprietary (A). Microsoft supplied MS-DOS (E).

- **The math:** Anchor Half-Life (t½) ≈ 1.5–2 years. Compaq and Columbia Data Products clean-room reverse-engineered the BIOS by 1983, pushing t½ down fast.
- **The result:** IBM fell into the Coupling Trap, spending capital to defend PC hardware even after the Diffusion Threshold. Microsoft, by contrast, let MS-DOS run on every PC clone, capturing VE(t) as it compounded.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 13, Issue 9, September 2026

DOI: 10.17148/IARJSET.2026.13918

3. Modern Edge Intelligence: IndoAI / Edge-AI Era (Compressed Half-Life)

IndoAI deploys physical EdgeBox units (A) in industrial environments, largely across India, to process AI tasks locally, paired with its Appization and Agent software (E).

- **The math:** Anchor Half-Life (t½) ≈ 2–3 years for a full reference design, compressing to a matter of months for the most commoditized chip components. Open-source AI models and cheap accelerator chips make copying hardware faster than ever.
- **The result:** because t½ is short, VA(t) collapses quickly. IndoAI avoids the Coupling Trap by building its software suite (E) to run on third-party cameras and sensors from day one, reaching Xc almost immediately.

Key Historical Comparison

Era	Anchor Asset (A)	Expansion Layer (E)	Anchor Half-Life (t½)	Winning Strategy
Mainframes	IBM S/360 machine	OS/360 & maintenance	~15–20 years	Proprietary hardware dominance
Personal Computers	IBM PC BIOS chip	MS-DOS / Windows OS	~1.5–2 years	Decouple software (Microsoft)
Edge AI	Reference hardware (EdgeBox)	NeurHub analytics & AI agents	~2–3 years	Immediate cross-platform expansion

The Strategic Takeaway

As technology paradigms accelerate, t½ shrinks. Hardware anchors are still essential for building early trust, but long-run value comes from decoupling software into an expansion layer before the hardware commoditizes.

4. CONCEPTUAL MODEL AND VISUALIZATIONS

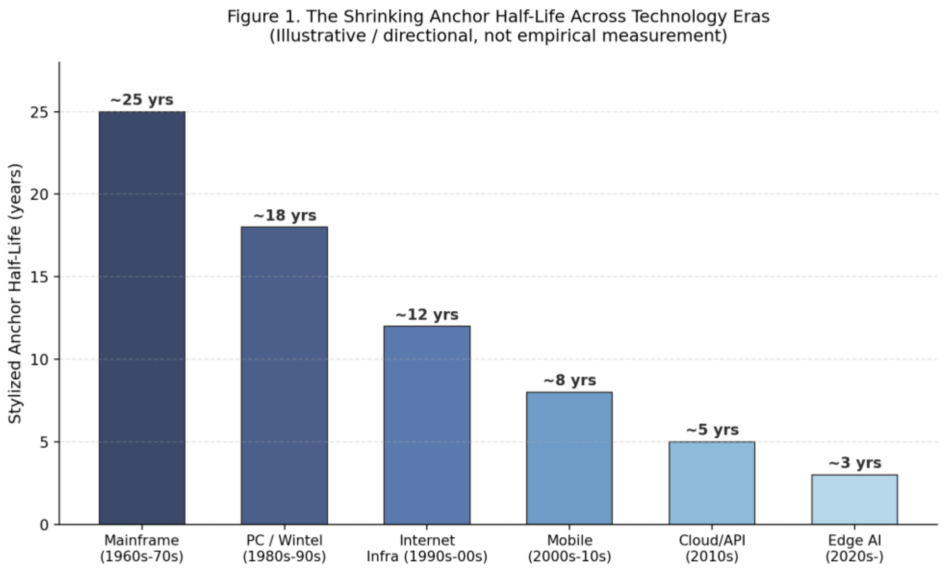

Figure 1. Compression of Anchor Half-Life Across Eras

Figure 1 lines up six broad technology eras and shows, roughly, how long each era's hardware anchor held its advantage before it was copied or standardized away:

- Mainframe (1960s–70s): held its edge for about a generation, roughly 25 years.
- PC / Wintel (1980s–90s): a reduced window of roughly 18 years.
- Internet Infrastructure (1990s–00s): down to roughly 12 years.
- Mobile (2000s–10s): down to roughly 8 years.
- Cloud/API (2010s): compressed further to roughly 5 years.
- Edge AI (2020s+): the shortest window yet, at roughly 3 years.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Impact Factor 8.066 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 13, Issue 9, September 2026
DOI: 10.17148/IARJSET.2026.13918

This bar chart supports Proposition 3 (P3). As open-source models, cheap accelerator chips, and global fabrication capacity proliferate, cloning a technology architecture gets faster, compressing the window of hardware insulation with each era.

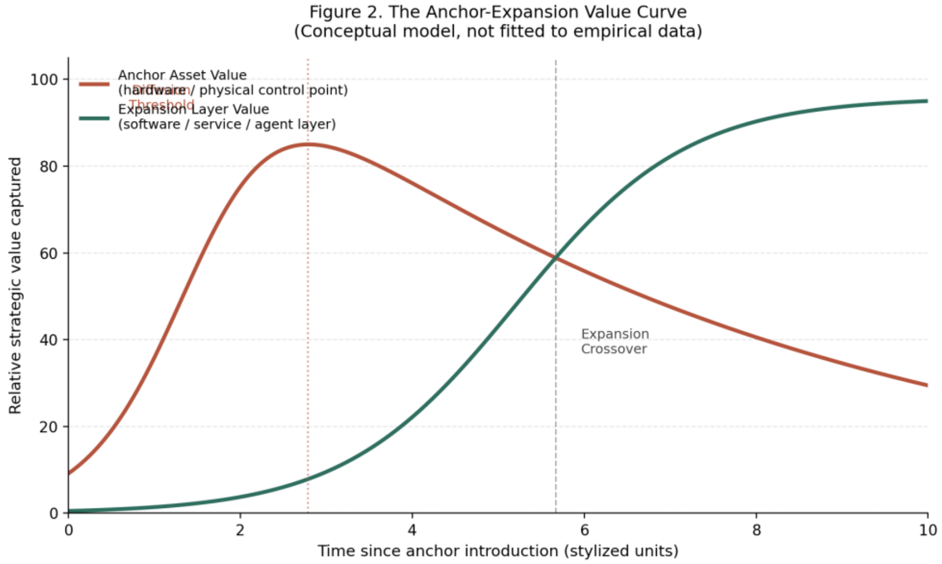

Figure 2. The Anchor-Expansion Value Curve

Figure 2 illustrates the Anchor-Expansion Value Curve over time, from a single firm's perspective:

- **Anchor Asset Value (red line):** represents hardware/physical control. It rises quickly during initial market entry to define standards, peaks at the Diffusion Threshold (D), and then fades as the category standardizes and commoditizes.
- **Expansion Layer Value (green line):** represents the software, service, or agent layer. It builds slowly at first while relying on the anchor for market credibility, then compounds rapidly after the Expansion Crossover, once it runs independently on third-party hardware.

The Expansion Crossover is the point where software/service value overtakes the anchor's physical value. Any firm running this playbook needs to plan for that crossover early, rather than reacting only after hardware margins erode (P2).

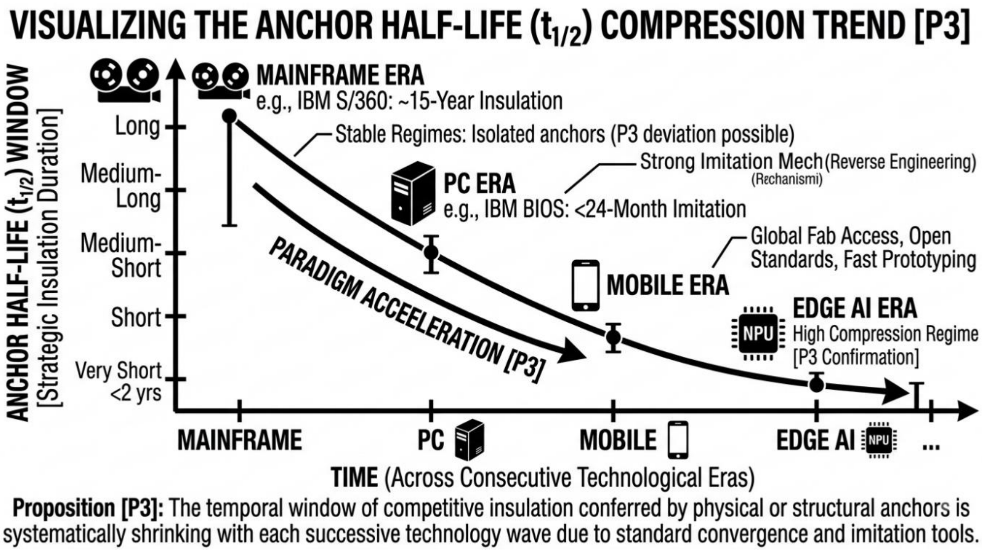

Figure 3. Visualizing the Anchor Half-Life (t½) Compression Trend (P3)

Figure 3 maps the strategic insulation window (t½) on the y-axis against consecutive technology eras on the x-axis:

- X-axis: tracks the shift from mainframes to PCs, mobile, and edge AI.
- Y-axis: tracks the half-life window, from “long” (15+ years) down to “very short” (roughly 1–2 years).

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⋇ Peer-reviewed & Refereed journal ⋇ Vol. 13, Issue 9, September 2026

DOI: 10.17148/IARJSET.2026.13918

The downward trajectory illustrates Proposition 3 (P3). Early eras allowed long insulation windows (IBM S/360 at roughly 15–20 years). By the PC era, clean-room reverse engineering had cut that down to under two years. In the edge-AI era, global fab access and open standards create a high-compression regime where physical anchors can decay in a matter of months. Strategic exceptions — "stable regimes" such as AWS Nitro or NVIDIA CUDA — deviate upward from the trend line, because high capital expenditure or deeply embedded developer workflows protect the anchor for longer.

5. CROSS-PARADIGM HISTORICAL CASE EXAMINATION

To examine how P1, P2, and P3 hold up in practice, we look at six distinct technology transitions spanning six decades.

5.1 Mainframes: IBM System/360 (1964)

IBM's System/360 is the quintessential hardware anchor. Before the S/360, software was tied to specific machine models, which created coordination failure across the industry. By internalizing architectural control and offering a scalable standard, IBM minimized asset-specificity risk for buyers (P1). Because technological uncertainty was high and capital barriers were steep, IBM enjoyed an extended Anchor Half-Life ($t½ \approx 15–20$ years). Its long-run value came from recognizing the Diffusion Threshold early: as hardware margins eroded, OS/360 and specialized services built compounding revenue streams that outlasted the initial hardware dominance (P2).

5.2 Personal Computing: IBM PC & BIOS Reverse Engineering (1981–1985)

IBM set out to define the PC standard on an open architecture, keeping only its BIOS proprietary. This drove massive early uptake (P1), but the isolating mechanism was weak. Clean-room reverse engineering by Columbia Data Products (1982) and Compaq (1983) commoditized the hardware anchor in under 24 months — a radical compression of t½ compared to the mainframe era (P3). IBM never decoupled its business model from the hardware; Microsoft and Intel, by contrast, treated their layers as portable expansions that thrived on IBM's own hardware clones, successfully navigating the Diffusion Threshold (P2).

5.3 Mobile Platforms: Apple vs. Google Android (2007–2010)

Apple pursued a vertically integrated strategy, using the iPhone as a closed anchor to retain end-to-end control. Google, in contrast, licensed Android horizontally as an open expansion layer from the start — a clear case of platform envelopment across diverse OEM hardware (P3). Android shows that an expansion layer can scale and spread faster than a single locked-down hardware anchor, illustrating the software-driven growth path described in P2.

5.4 Cloud Infrastructure: AWS Nitro & Custom Silicon (2015–Present)

Amazon Web Services acquired Annapurna Labs in 2015 to develop specialized silicon — Nitro hypervisor-offload cards and Graviton CPUs. These hardware anchors aren't sold as standalone products; they exist purely to optimize performance, security, and cost for AWS's core expansion layer, its public cloud services. High capital requirements, complex engineering cycles, and tight integration into a massive service layer have insulated this anchor, giving it an extended t½ that stands as a capital-backed exception to P3.

5.5 Software-Defined Anchors: NVIDIA CUDA (2006–Present)

NVIDIA's ecosystem shows that an anchor doesn't have to be physical. Compute Unified Device Architecture (CUDA) is a software API that became the de facto standard for parallel GPU programming. By building developer workflows around CUDA, NVIDIA made switching costs prohibitive. CUDA functions as an interface-level anchor with a multi-decade half-life (t½ of roughly 18–20 years), showing that a software architecture can be an even stronger isolating mechanism than physical hardware (a P3 exception).

5.6 Edge Artificial Intelligence: High-Compression Regimes (Current Paradigm)

The modern edge-AI landscape, driven by localized inferencing and NPU acceleration, represents the fastest standard convergence and anchor compression seen yet. Open-source model optimization, low-cost accelerator chips, and global fabrication access have structurally minimized hardware differentiation, cutting the window of proprietary hardware advantage down to a few years at most, and to just months for the most commoditized components.

- **Case illustration (IndoAI / Indian enterprise context):** in developing markets like India, where infrastructure constraints — bandwidth, power reliability — make edge computing a necessity, IndoAI uses its EdgeBox hardware as an initial anchor to validate performance, reduce transaction costs and establish credibility (P1). IndoAI's software suite — Appization, *app.Indo.ai*, and *agents.Indo.ai* — is engineered for cross-platform compatibility across third-party camera systems from day one. This proactive decoupling recognizes how fast

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 13, Issue 9, September 2026

DOI: 10.17148/IARJSET.2026.13918

half-life decays in this era (P3) and lets IndoAI avoid the coupling trap, maintaining compounding service value long after EdgeBox itself commoditizes (P2).

The architecture behind this case is documented in more technical detail elsewhere by the author and colleagues: the EDGEBENCH validation framework [27] formalizes the 225-parameter benchmarking used to establish EdgeBox's initial credibility as an anchor, while a companion paper on SLM-governed edge-AI marketplaces [28] documents the Appization software layer's design and evaluation. A peer-reviewed paper on vision-language models as a natural-language interface layer for this same tool suite [21] was accepted in 2026 and provides further detail on the expansion layer's design-verify pipeline referenced in this section.

6. COMPARATIVE CASE MAPPING

Paradigm & Firm	P1: Architectural Uncertainty & Anchor Value	P2: Decoupling & Expansion Value Capture	P3: Half-Life Compression (t½) & Exceptions
Mainframe (IBM S/360)	Owns full-system integration to resolve high initial market ambiguity.	Captures value through OS/360 and services over an extended period.	Stable regime: t½ ≈ 15–20 years, driven by high capital intensity.
Personal Computer (IBM vs. Wintel)	IBM uses a proprietary BIOS to establish the early PC standard.	Microsoft and Intel capture long-run surplus by decoupling software/CPU from IBM.	Rapid compression: clean-room reverse engineering commoditizes the BIOS in under 24 months.
Mobile (Apple vs. Android)	Apple uses iPhone hardware to control the end-to-end smartphone experience.	Google's Android platform envelops the market through horizontal deployment.	Accelerated scale: Android reaches scale faster than any single hardware line.
Cloud Infrastructure (AWS Nitro)	Custom silicon optimizes hypervisor security and throughput.	Custom hardware directly protects and lowers costs for AWS's cloud/SaaS layer.	Strategic exception: high CapEx and complex silicon extend t½ despite the industry trend.
Software-Defined (NVIDIA CUDA)	Standardizes GPU parallel-programming interfaces for researchers.	Developer lock-in drives high-margin enterprise software and chip sales.	Software exception: interface switching costs extend t½ to roughly 18–20 years.
Edge AI (IndoAI context)	EdgeBox hardware establishes initial field reliability in India.	Appization and Agents.Indo.ai run on third-party cameras from Day 1.	Maximum-compression regime: hardware edge decays in a few years at most; immediate software decoupling required.

7. BOUNDARY CONDITIONS & STRATEGIC LIMITATIONS

The Anchor-and-Expand framework has three key limitations worth flagging:

1. **Post-Standardization Regimes:** deploying a proprietary anchor after an industry has already reached its dominant design [22] tends to destroy value, since the market resists fragmentation at that stage.
2. **Capital Asymmetry:** early-stage firms that try to field custom hardware anchors face high burn rates. Outsourcing anchor requirements through strategic alliances or reference platforms is usually preferable when appropriability regimes are weak.
3. **Interface Lock-in Exceptions:** anchors that entrench complex developer workflows (CUDA) or that rest on high capital-expenditure barriers (AWS Nitro) resist rapid half-life decay — proof that structural lock-in can counter the general compression trend.

IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 13, Issue 9, September 2026

DOI: 10.17148/IARJSET.2026.13918

8. CONCLUSION AND RESEARCH AGENDA

The Anchor-and-Expand Principle offers a simple way to think about a recurring problem in technology strategy: **owning the anchor can create early advantage, but owning it for too long can become a constraint.** The framework brings together four elements — Anchor Asset (A), Expansion Layer (E), Diffusion Threshold (D), and Anchor Half-Life (t½) — to explain when firms should establish control and when they should start moving value creation toward a more portable software or service layer.

The historical cases suggest the period of protection created by a technological anchor varies widely. Mainframes, PC hardware, mobile platforms, cloud infrastructure, and software-defined ecosystems each show a different form of strategic insulation. AWS Nitro and NVIDIA CUDA also show that anchor advantage doesn't always decay quickly — strong capital requirements, technical complexity, or deeply embedded developer workflows can extend its life well beyond the general trend.

The central implication isn't that firms should avoid hardware or physical control. Rather, **they should know when to stop depending on it.** The strategic advantage increasingly lies in building the expansion layer early enough that it can survive the anchor's commoditization.

The framework remains conceptual, and the t½ estimates presented here are illustrative rather than empirically measured. Future research could test these propositions against larger cross-industry datasets, develop more rigorous measures of anchor decay, and examine whether the same pattern holds outside computing and technology markets.

The natural next steps follow directly from those limitations: building a cross-industry dataset large enough to estimate t½ empirically rather than illustratively; testing whether the compression trend in Proposition 3 holds outside computing; and extending the model's treatment of D from a fixed threshold to a probabilistic one, reflecting how standardization actually diffuses across regions and regulatory regimes. Until then, the Anchor-and-Expand model should be treated as what it is: a structured way of asking when — not merely whether — owning the anchor still pays.

REFERENCES

[1]. Schmalensee, R. (2015). Pricing the razor: A note on two-part tariffs. International Journal of Industrial Organization, 42, 19–22.
[2]. Wolf, C. (2026). The making and unmaking of uneven development: The role of anchor firms in creating and overcoming industrial decline in East Germany. Journal of Economic Geography, 26(3), 375–397. https://doi.org/10.1093/jeg/lbaf055
[3]. Coase, R. H. (1937). The nature of the firm. Economica, 4(16), 386–405.
[4]. Williamson, O. E. (1975). Markets and Hierarchies: Analysis and Antitrust Implications. Free Press.
[5]. Williamson, O. E. (1985). The Economic Institutions of Capitalism. Free Press.
[6]. Klein, B., Crawford, R. G., & Alchian, A. A. (1978). Vertical integration, appropriable rents, and the competitive contracting process. The Journal of Law and Economics, 21(2), 297–326.
[7]. Pies, I., Beckmann, M., & Hielscher, S. (2010). Value creation, management competencies and global corporate citizenship: An ordonomic approach to business ethics in the age of globalization. Journal of Business Ethics, 94, 265–278. https://doi.org/10.1007/s10551-009-0263-1
[8]. Teece, D. J. (1986). Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. Research Policy, 15(6), 285–305.
[9]. Teece, D. J. (2006). Reflections on “Profiting from Technological Innovation”: Issues and implications for research and practice. Research Policy, 35(8), 1131–1146.
[10]. Teece, D. J. (2018). Profiting from innovation in the digital economy: Enabling technologies, standards, and licensing models in the wireless world. Research Policy, 47(8), 1367–1387.
[11]. Barney, J. (1991). Firm resources and sustained competitive advantage. Journal of Management, 17(1), 99–120.
[12]. Peteraf, M. A. (1993). The cornerstones of competitive advantage: A resource-based view. Strategic Management Journal, 14(3), 179–191.
[13]. Rumelt, R. P. (1984). Towards a strategic theory of the firm. In R. Lamb (Ed.), Competitive Strategic Management (pp. 556–570). Prentice-Hall.
[14]. Rochet, J.-C., & Tirole, J. (2003). Platform competition in two-sided markets. Journal of the European Economic Association, 1(4), 990–1029.
[15]. Eisenmann, T., Parker, G., & Van Alstyne, M. (2011). Platform envelopment. Strategic Management Journal, 32(12), 1270–1285.

DOI: 10.17148/IARJSET.2026.13918

[16]. Gawer, A., & Cusumano, M. A. (2002). Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation. Harvard Business School Press.

[17]. Iansiti, M., & Levien, R. (2004). The Keystone Advantage: What the New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability. Harvard Business School Press.

[18]. Moore, J. F. (1993). Predators and prey: A new ecology of competition. Harvard Business Review, 71(3), 75–86.

[19]. Gujar, Vivek. (2026). The Camera Marketplace as a Certified Outcome Product: Integrating the 10Ps Framework, DUAL Architecture and Growth Hacking Strategy for e Marketplace, Indian Journal of Computer Science and Technology, Volume 05, Issue 01 (Jan-April 2026), PP: 684-695, DOI: 10.59256/indjcst.20260501082

[20]. Gujar, V. (2025). Marketing Mix 3.0: Personalization, platforms and performance analytics in the algorithmic era of growth hacking — A review. International Advanced Research Journal in Science, Engineering and Technology, 12(12). https://doi.org/10.17148/IARJSET.2025.121203

[21]. Gujar, V., Gujar, A., & Rathore, A. K. (2026). Vision language models as the reasoning layer of an AI-CCTV planning toolkit: A case study of IndoAI's calculator suite. Global Journal of Computer Science and Technology, Ahead of Print, 1–15. https://doi.org/10.34257/GJCSTD259174

[22]. Utterback, J. M., & Abernathy, W. J. (1975). A dynamic model of process and product innovation. Omega, 3(6), 639–656.

[23]. Henderson, R. M., & Clark, K. B. (1990). Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. Administrative Science Quarterly, 35(1), 9–30.

[24]. Teece, D. J., Pisano, G., & Shuen, A. (1997). Dynamic capabilities and strategic management. Strategic Management Journal, 18(7), 509–533.

[25]. Christensen, C. M. (1997). The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail. Harvard Business School Press.

[26]. Christensen, C. M., & Rosenbloom, R. S. (1995). Explaining the attacker's advantage: Technological paradigms, organizational dynamics, and the value network. Research Policy, 24(2), 233–257.

[27]. IndoAI Technologies Pvt. Ltd. (Internal Technical Documentation). EDGEBENCH: A 225-parameter validation framework for edge camera benchmarking. Unpublished internal document.

[28]. Gujar, V., et al. (in preparation). Appization and SLM-governed edge AI marketplaces. Manuscript in preparation.