

## Non-peer-reviewed Articles

## Governing Through Time: A Theoretical Framework for Platformer-Defined Sovereignty in the Mobility Industry

Akio Tokuda \*

### Abstract:

This paper develops a theoretical framework for analysing the governance regime through which platformers exercise sovereignty over the mobility industry. The framework, which I call *Platformer Defined Sovereignty* (PDS), reorganises the analysis of platform competition around the architecture of time: the cadence at which updates flow, the protocol through which integration is governed, and the temporal coordination across which value is captured. As Software Defined Vehicles (SDV) proliferate and Physical AI enters the mobility stack, the question of who governs the temporal infrastructure — not merely what software runs — becomes the decisive strategic variable. Drawing on the thirty-year history of the mobile telecommunications industry as a completed natural experiment, the paper constructs an analytical toolkit comprising four governance mechanisms (G), four temporal disciplines (T), four governance handles (H), and three revenue functions (V). The toolkit yields a master expression —  $Profit = f(What, When, How) \rightarrow Where$  — through which any platformer's sovereignty regime can be diagnosed. The paper then develops the *CPS Value Emergence Matrix*, which reorganises platform governance around two independent boundaries (function and meaning), identifying the Q1  $\rightarrow$  Q2 horizontal connection as the mechanism by which user-generated value enters coordinative recombination. The *Three-Loop Model* gives institutional form to multi-stakeholder coordination across heterogeneous time scales — Polity-in-the-Loop, Alliance-in-the-Loop, and Machine-in-the-Loop — distinguishing autonomation (*jidoka*) from mere automation. The paper closes with the prescriptive concept of the *receptive platform* governed by *juseki* (responsibility-taking), which dissolves the structural misalignment between operational sovereignty (held by platformers) and juridical sovereignty (borne by OEMs). The framework's location relative to the digital-colonisation literature is registered, and its implications for Physical AI governance are identified as a forward research direction.

### Keywords:

platformer governance; mobility industry; temporal discipline; software defined vehicle (SDV); Physical AI; cyber-physical systems; sovereignty

---

\* Professor, College of Business Administration, Ritsumeikan University, Academic Visitor, Saïd Business School, University of Oxford

## Table of Contents

1. Introduction
2. The Strategic Space: A Typology of Mobility Platformers
  - 2.1 Two Axes
  - 2.2 Four Types
  - 2.3 Integration versus Binding
3. The Analytical Toolkit
  - 3.1 Four Governance Mechanisms (G)
  - 3.2 Four Temporal Disciplines (T)
  - 3.3 A Note on Relational Time
  - 3.4 Four Governance Handles (H)
  - 3.5 Three Revenue Functions and the Master Expression
  - 3.6 Scope Conditions: The Toolkit as Structural Propositions
  - 3.7 Falsification Anchors: When the Framework Fails
4. The CPS Value Emergence Matrix and the Three-Loop Model
  - 4.1 The CPS Value Emergence Matrix
  - 4.2 The Q1 → Q2 Horizontal Connection
  - 4.3 Theoretical Inheritance: Two Independent Uncertainties
  - 4.4 The Three-Loop Model
  - 4.5 Alliance-in-the-Loop, Not Human-in-the-Loop
  - 4.6 Automation versus *Jidoka* (Autonomation)
5. The Receptive Platform and *Juseki* Governance
  - 5.1 Governance Without Responsibility
  - 5.2 The Receptive Platform
  - 5.3 Undefined-by-Design
  - 5.4 Full Self-Driving as a Strong Promotive Condition for Q2
6. Conclusion: Implications for Digital Colonisation

## 1. Introduction

Who defines the automobile? For more than a century, the answer seemed self-evident: the carmaker who designed it, built it, and sold it. The product's specifications and its evolutionary trajectory rested in the manufacturer's hands. That self-evidence is now dissolving. Tesla, BYD, Google, and Huawei have entered the mobility industry not as carmakers but as designers of the architecture that governs how mobility is experienced. The legal duty for manufacture still rests with the original equipment manufacturer (OEM), but the power to define what an automobile is — and what it may become — no longer does.

This paper takes that displacement seriously. I argue that the contest now unfolding in the mobility industry is not, at its core, a contest of software. It is a contest over the architecture of time: who sets the tempo of updates, who controls the cadence of integration, who governs the rhythm at which value flows back to its source. The decisive variable is not what runs on the vehicle but who holds the conditional-setting rights — over compatibility tests, application programming interfaces, update discipline, distribution channels, and certification regimes — through which the vehicle's behaviour is, day by day, redefined.

The framework I develop calls this constellation of conditional-setting rights *Platformer Defined Sovereignty* (PDS) — a deliberate echo of the more familiar SDV (Software Defined Vehicle) and PDV (Platformer Defined Vehicle) but distinct from both. SDV captures the technological substrate. PDV captures the redefinition of the vehicle as such. PDS captures the *sovereignty regime* — the architecture by which a platformer governs the temporal and spatial flow of value within a mobility ecosystem.

A clarification on the use of *sovereignty* is needed before the framework is developed. In conventional political theory, sovereignty bundles four properties: final decision authority, exclusion rights, enforcement capacity, and legitimacy. The platformer-economy configuration that this paper analyses has, structurally, *separated* these properties. I therefore distinguish two layers of sovereignty whose alignment is the central analytical concern. *Operational sovereignty* refers to the daily exercise of conditional-setting rights — over specifications, updates, compatibility, distribution, and certification. This is the layer that CDD/CTS/GMS, OTA discipline, and API governance operate. *Juridical sovereignty* refers to the final-instance authority over responsibility, liability, and legality — the layer through which physical-safety obligations, product-liability law, and criminal accountability are exercised. In the conventional industrial regime, these two layers coincided within the OEM. In the platformer-economy configuration, they have come apart: the platformer holds operational sovereignty over the vehicle's behaviour while the OEM continues to bear juridical sovereignty over physical consequences. The configuration that Section 5 analyses as *governance without responsibility* is structurally this misalignment. The framework's prescriptive task — developed in Section 5 as the *receptive platform* — is the reconstruction of an institutional architecture in which the two layers re-align, not within a single firm but across a coordinated consortium.

The paper is structured in five sections following this introduction. Section 2 maps the strategic space of mobility platformers through a typology along two axes — OS governance holding and vertical-integration versus network. Section 3 constructs the analytical toolkit: the four governance mechanisms (G), four temporal disciplines (T), four governance handles (H), and three revenue functions (V). Section 4 develops the CPS Value Emergence Matrix and the Three-Loop Model as the framework's prescriptive architecture. Section 5 introduces the concept of the *receptive platform* governed by *juseki* (responsibility-taking). Section 6 concludes by locating the framework's contribution relative to the digital-colonisation literature.

The empirical application of this framework to six specific OEMs (Toyota, Honda, Hyundai, Volkswagen, BMW, Mercedes-Benz) and four specific platformers (Tesla, BYD, Google, Huawei) is developed separately.

## 2. The Strategic Space: A Typology of Mobility Platformers

### 2.1 Two Axes

The strategic space of mobility platformers is constructed along two axes:

- *Axis 1*: OS governance holding (yes / no)
- *Axis 2*: Vertical integration versus network

The first axis requires careful definition. OS governance is not the same as legal ownership of operating system code. Google does not own the Linux kernel that runs at the base of Android; the kernel is licensed under the GPL and modified by countless OEMs. Yet Google holds *governance* over the Android ecosystem through three operational instruments: the Compatibility Definition Document (CDD) which specifies conformance requirements, the Compatibility Test Suite (CTS) which provides the machine-readable enforcement, and the Google Mobile Services (GMS) license which provides the commercial enforcement. The combination produces governance without ownership — and it is governance, not ownership, that constitutes the strategic variable.

The second axis distinguishes two coordinative forms. *Vertical integration* organises supply, production, and distribution under unified responsibility within a firm boundary. *Network* organises the same functions across multiple firms through interface conditions. The two axes generate four types of mobility platformer.

### 2.2 Four Types

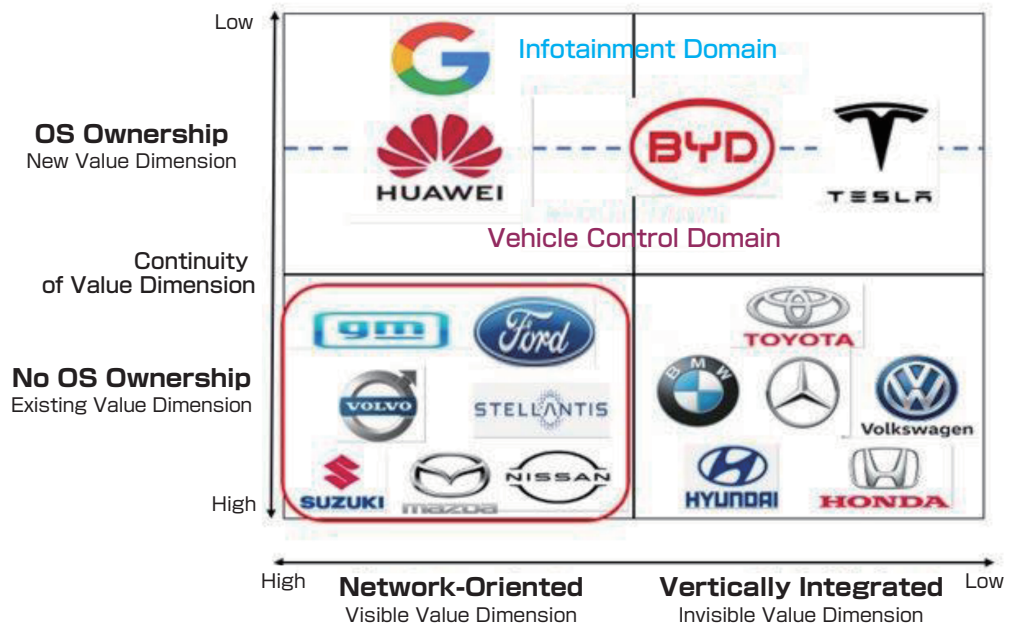
**Type 1 (OS holding × Vertical integration):** Tesla, BYD. Closed integration of hardware, OS, and services. Strong vertical control with internal definition of governance conditions.

**Type 2 (OS holding × Network):** Google, Huawei. Governance over networked OEMs through binding conditions on interface specifications. The platformer does not produce vehicles directly but governs the conditions under which vehicles operate.

**Type 3 (OS non-holding × Vertical integration):** Toyota, Honda, Hyundai, Volkswagen, BMW, Mercedes-Benz. Historically established OEMs that organise production through vertical integration but do not hold OS governance. The strategic question facing these firms — how to retain residual control rights over the vehicle without holding OS governance — constitutes the central empirical question for the mobility industry's sovereignty transition.

**Type 4 (OS non-holding × Network):** Internally heterogeneous. Three sub-types: - *4α (Adaptive OEMs)*: GM, Ford, Stellantis, Nissan, Mazda, Suzuki, Volvo. Firms that adapt to externally provided platforms while differentiating through complementary capabilities. - *4β (Technical-Enabler Tier 1s)*: Bosch, Denso, ZF, Continental, DXC Technology. Suppliers and software integrators that fill the integration gap; what I call *manufacturing platformers* who hold physical-layer governance without holding code-layer governance. - *4γ (Standardisation Consortia)*: AUTOSAR, COVESA, SOAFEE, Eclipse SDV, Zephyr,

OpenSDV. Industry-wide consortia that define the protocol layer.



**Figure 1.** *The Strategic Space of Mobility Platformers: A typology along OS governance holding × Vertical integration / Network<sup>1)</sup>*

Source: author.

### 2.3 Integration versus Binding

A critical distinction within Type 2 deserves explicit registration. This paper argues that platformer sovereignty operates through two structurally different mechanisms:

- *Integration*: internal consolidation under unified responsibility.
- *Binding*: governance over external resources through interface conditions.

Google's governance over the Android ecosystem operates through binding, not integration. The conditional-setting rights bound to Android Automotive OS (AAOS) reach far beyond infotainment — extending into the Vehicle Hardware Abstraction Layer (which determines what data crosses the boundary between OS and physical vehicle), the data layer (whose collection and retention are governed by AAOS update policy), application distribution (which review and signing procedures are bound to the CTS), and security and key management (whose upstream specifications govern downstream OEM implementation). Each of these is, in itself, a legitimate technical concern. Together, they define what binding governance looks like in operational terms — and it is the same structural form that the consumer-information platforms deployed in their earlier industries, now extended into a regulated physical industry.

### 3. The Analytical Toolkit

#### 3.1 Four Governance Mechanisms (G)

The first instrument measures the static structural capacity of a platformer's governance — the structural force it commands, not the speed at which it acts. Four mechanisms are distinguished, each indexed to an empirical type from the mobile-industry case.

**G<sub>1</sub> — Vertical penetrating force** (Apple's metric). The capacity to enforce design discretion across the full vertical stack. Formally:  $G_1 = (\sum_j \alpha_j) \cdot (1 / L\_total)$ , where  $\alpha_j$  is the strength of unilateral design discretion at layer  $j$ , and  $L\_total$  is the total number of layers.

**G<sub>2</sub> — Horizontal expansive force** (Google's metric). The reach of a common rule-set across devices, OEMs, and complementor firms. Formally:  $G_2 = (\sum_k \beta_k) \cdot (1 / A\_total)$ , where  $\beta_k$  is the compliance rate of external actors with API  $k$ .

**G<sub>3</sub> — Collective binding force** (Nokia's metric, and the framework's signature variable). The collective resilience of a consortium-type governance regime. Formally:  $G_3 = (T\_life / R\_deviance) \times (\sum_i \gamma_i \cdot \chi)$ , where  $\gamma_i$  is the willingness of member firm  $i$  to suppress its own design preferences in favour of the common rule. The Symbian alliance failed because  $\gamma_i$  declined and  $\chi$  became inoperative.

**G<sub>4</sub> — External negotiation force** (Samsung's metric). The leverage a non-OS-holding manufacturer commands by virtue of its market share and core-component holdings. Formally:  $G_4 = (MS \times C\_core) / D\_OS$ .

The four mechanisms map onto the four empirical types in a non-arbitrary way. The typology and the toolkit are co-designed.

#### 3.2 Four Temporal Disciplines (T)

Static governance capacity (G) must be activated by temporal discipline. Four disciplines correspond to the four governance mechanisms:

**T<sub>1</sub> — Update pulse strength** (the Apple beat).  $T_1 = A\_amp \cdot \omega\_reg$ , where  $A\_amp$  is the amplitude of a single update event and  $\omega\_reg$  is the regularity of the update clock.

**T<sub>2</sub> — Synchronisation rate** (the Google sync).  $T_2 = s(t) \propto (H\_c \cdot H\_b) / N\_frag$ , where  $s(t)$  is the proportion of endpoints running current specification.

**T<sub>3</sub> — Integration success rate** (the Nokia variable, the analytical key to consortium failure and consortium possibility).  $T_3 = p\_eff \approx (p + \Delta p)^{A_4}$ , where  $\Delta p = f(H\_a, H\_c)$ . The crucial property of  $T_3$  is its *exponential* dependence on coupling: as  $A_4$  rises, small drops in  $p$  compound multiplicatively. In consortium governance, time is governed not by speed but by certainty.

**T<sub>4</sub> — Preemption velocity** (the Samsung edge).  $T_4 = v\_preempt$ , where  $\tau\_self \ll \tau\_ext$ . The time-difference rent  $\Delta\tau = \tau\_ext - \tau\_self$  is the economic substance of  $T_4$ .

**Figure 2.** *The G–T Correspondence: Four Governance Archetypes from the Mobile Industry*

| Strategic Type | G (Static Capacity)                 | T (Temporal Discipline)                 | H (Principal Handles)           | Outcome           |
|----------------|-------------------------------------|-----------------------------------------|---------------------------------|-------------------|
| Apple          | G <sub>1</sub> vertical penetrating | T <sub>1</sub> update pulse             | H <sub>a</sub> × H <sub>b</sub> | Success           |
| Google         | G <sub>2</sub> horizontal expansive | T <sub>2</sub> synchronisation rate     | H <sub>c</sub> × H <sub>b</sub> | Success           |
| Samsung        | G <sub>4</sub> external negotiation | T <sub>4</sub> preemption velocity      | H <sub>c</sub> × H <sub>b</sub> | Success           |
| Nokia          | G <sub>3</sub> collective binding   | T <sub>3</sub> integration success rate | H <sub>a</sub> × H <sub>c</sub> | Failure (Symbian) |

**Footnote:** Nokia's G<sub>3</sub> is the variable on which next-generation consortium governance depends. The framework treats Nokia not as a closed failure but as the prototype whose paired T<sub>3</sub> requires redesign.

### 3.3 A Note on Relational Time

T<sub>1</sub>–T<sub>4</sub> are not measurements of time in the Newtonian sense. They are not indices of how fast a firm moves, how long it sustains advantage, or when it enters a market. The strategic-management tradition has examined time in those three senses — timing (Lieberman & Montgomery, 1988), persistence (Barney, 1991), speed (Eisenhardt, 1989; Teece, 2007) — and all three presuppose absolute time: a universal container against which each firm's choices are measured.

T<sub>1</sub>–T<sub>4</sub> presuppose something different. The temporal disciplines describe *relational time* — a rhythm that arises *between* governed subjects, not within a single subject. Apple's annual pulse exists not because Apple decides to update once a year, but because the relationship between Apple, its developers, its users, and its supply chain *becomes* an annual rhythm when the governance mechanism G<sub>1</sub> is operated through the handles H<sub>a</sub> and H<sub>b</sub>. T<sub>1</sub> is the time that emerges *as* the governance regime; it does not exist apart from the regime.

The implication: I am not claiming that platformers move faster than incumbents. I am claiming that they *constitute the time* in which their ecosystems move. This is a different kind of strategic-management theory from the one that asks “how fast” or “when.” It is closer in spirit to phenomenological accounts of inter-subjective time, though I make no philosophical commitment beyond the empirical observation that platformer-defined time, once instituted, governs the temporal conduct of all participants in the ecosystem.

*Observational anchors for relational time.* The concept becomes empirically tractable when its operational signatures are specified. Three observable markers indicate that a platformer has successfully instituted relational time within its ecosystem: (i) *cycle entrainment* — the proportion of ecosystem participants whose internal release cycles have synchronised to the platformer's clock (Apple's annual September event, Google's monthly Android security bulletin, semiconductor industry's foundry process roadmap); (ii) *deviation cost* — the observable penalty borne by participants who fail to align with the platformer's tempo (delayed app reviews, lost compatibility certification, lagged GMS access); and (iii) *forecasting orientation* — the degree to which ecosystem participants' planning documents reference the platformer's cycle rather than industry-internal cycles (an OEM strategy document that organises its three-year plan around Android version trajectories rather than its own model-year cycle has lost temporal sovereignty in the framework's terms).

Relational time is not metaphysical; it is observable through these three signatures. *Where these signatures are absent, the framework predicts that no relational time has been instituted, regardless of how fast the candidate platformer moves.*

### 3.4 Four Governance Handles (H)

Capability (G) and cadence (T) are activated through four operational handles. The hydraulic metaphor that organises the framework locates each handle in a water-system component:

- **H\_a — Conformance gate** (intake weir): CDD/CTS, App Review. Filters out impurities before water enters the system.
- **H\_b — Distribution canal** (canal): OTA, FOTA, store distribution. Delivers approved components to all endpoints.
- **H\_c — Runtime levee** (levee): OS sandbox, API permissions, isolation. Contains runtime behaviour within prescribed bounds.
- **H\_d — Capture waterwheel** (waterwheel): in-app payment, advertising identifiers, analytics platforms. Converts the flow into economic value.

A critical asymmetry: H\_d is the only handle that is *not* a primary determinant of any temporal discipline. H\_d appears in the  $H \times T$  influence matrix as a supporting indicator across all four T disciplines, never as a primary effect. Yet H\_d is the principal in the revenue functions developed below. *The wheel turns where the water has been channelled, not where the water is channelled.* This asymmetry — H\_d as supporting in time control, principal in value capture — is the mathematical statement of the framework's central operational distinction: governance is the conversion of spatial structure (H\_a, H\_b, H\_c) into temporal discipline (T<sub>1</sub>–T<sub>4</sub>), which in turn delivers value to the capture point (H\_d).

|                  | Apple Type (Sky Lake and Kegan Falls)                                                                               | Google Type (The Mainstream of the Tone River)                                      | Samsung Type (Lake Kasumigaura and Its Canal Network)                                                          |
|------------------|---------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
|                  |                                  |  |                            |
| Strategy         | Vertical penetration Capability (G1)<br>× Annual Pulse (T1)                                                         | Horizontal Expansion Capability (G2)<br>× Constant Synchronization (T2)             | External Bargaining Power (G4)<br>× Preemptive Speed (T4)                                                      |
| Monetize         | Premium (drop height) × fee (waterwheel).<br>Forces a market reset through a high degree of proprietary discretion. | Tolerates riverbed mud (fragmentation) and earns from surface ad flow (volume).     | While staying subordinate to the mainstream, implements its own special zone early and harvests time-lag rent. |
| Revenue Function | $V1 \propto \text{Premium} \times T1$                                                                               | $V2 = k2 \times G2 \times T2 \times (1-F)$                                          | $V3 = \text{Share} \times \text{Margin} \times \max(0, \Delta\tau)$                                            |

**Figure 3.** Three Water-System Models of Platformer Governance: Apple, Google, and Samsung as hydraulic architectures

Source: author.

### 3.5 Three Revenue Functions and the Master Expression

The temporal disciplines, when paired with their corresponding G and operationalised through their H combinations, deliver three distinct revenue functions:

**V<sub>1</sub> (Apple type — height-and-pulse capture):**  $V_1 \propto \text{Premium} \times T_1(\text{Pulse})$

**V<sub>2</sub> (Google type — flow-and-synchronisation capture):**  $V_2 = k_2 \cdot G_2(\text{Scale}) \cdot T_2(\text{Sync}) \cdot (1 - F)$ , where F is fragmentation friction.

**V<sub>3</sub> (Samsung type — time-differential rent extraction):**  $V_3 \propto \text{Share} \cdot \text{Margin} \cdot \max(0, \Delta\tau)$

The three functions are not normative ideals; they are comparative diagnostic instruments. A firm's PDS rarely fits one function cleanly. Empirical analysis demonstrates that Tesla operates a hybrid of V<sub>1</sub> and a fleet-internal special case of V<sub>2</sub>, and that BYD operates a dual-mode configuration whose revenue function combines depth in the physical layer with breadth in the software layer.

The framework closes with the master expression that ties the toolkit together:

$$\text{Profit} = f(\text{What, When, How}) \rightarrow \text{Where}$$

The arrow denotes the operational forced convergence of the value flow to the capture point. *What* (territorial scope) and *When* (temporal cadence) are bound together through *How* (protocol) and forced to converge at *Where* (revenue point). The platformer who controls the script of this convergence is sovereign; the firm that participates without controlling the script is governed.

**Figure 4.** The  $H \times T$  Influence Matrix: How four governance handles produce four temporal disciplines

|                                        | T <sub>1</sub><br>(update pulse) | T <sub>2</sub><br>(synchronisation) | T <sub>3</sub><br>(integration success) | T <sub>4</sub><br>(preemption)                             |
|----------------------------------------|----------------------------------|-------------------------------------|-----------------------------------------|------------------------------------------------------------|
| H <sub>a</sub><br>(conformance gate)   | ● enforces $\omega_{\text{reg}}$ | ○ certified versions sync           | ● lifts p (filters impurity)            | ○ proprietary cert shortens $\tau_{\text{self}}$           |
| H <sub>b</sub><br>(distribution canal) | ● generates A <sub>amp</sub>     | ● drives s(t)                       | ○ staged delivery contains failure      | ● proprietary distribution shortens $\tau_{\text{self}}$   |
| H <sub>c</sub><br>(runtime levee)      | ○ rollback capacity              | ● abstraction drives s(t)           | ● corrects vulnerability ( $\Delta p$ ) | ● proprietary implementation shortens $\tau_{\text{self}}$ |
| H <sub>d</sub><br>(capture waterwheel) | ○ effective measurement          | ○ divergence detection              | ○ failure event tracking                | ○ preemption-effect measurement                            |

**Footnote:** H<sub>d</sub> appears as supporting in every column and principal in none — the formal statement that the capture waterwheel does not produce time, it harvests value from time produced by the other three handles. The asymmetry distinguishes time control (H<sub>a</sub>, H<sub>b</sub>, H<sub>c</sub>) from value capture (H<sub>d</sub>).

### 3.6 Scope Conditions: The Toolkit as Structural Propositions

A clarification on the analytical status of the expressions presented in this section. The G, T, and V formulas are *not* econometric estimation models, and the dimensionless indices (A<sub>4</sub>,  $\gamma_{\text{BOM}}$ , s(t),  $\Delta\tau$ ) are not direct empirical measurements. They are *structural propositions* — formal statements of how the variables relate under specified scope conditions. Each

formula carries within it the conditions under which it holds, and identifying these conditions is, I would argue, more important for the framework's evaluation than estimating numerical values.

The principal scope conditions are as follows:

- $G_2 = (\sum_k \beta_k) \cdot (1/A_{\text{total}})$  assumes that APIs  $k$  are weighted by their governance-criticality. A naive count that treats all APIs equally — peripheral APIs equivalent to core APIs — produces a measure that does not reflect actual platform governance. The proposition holds when  $\beta_k$  is operationalised as governance-critical compliance, not as gross API conformance.
- $T_3 = p_{\text{eff}} \approx p^A(A_4)$  assumes mutual interdependence of modules and the failure of localised failure-containment. The exponential dependence on coupling that the formula expresses is the structural feature that distinguishes consortium governance from federated governance, but it requires that failures cannot be confined to single modules. When isolation succeeds (when  $H_c$  levees hold), the exponent  $A_4$  effectively drops, and the formula's distinctive prediction (catastrophic compounding) does not obtain. The Symbian failure was, in this sense, a failure of isolation as much as a failure of integration.
- $T_4 = v_{\text{preempt}}$ , where  $\tau_{\text{self}} \ll \tau_{\text{ext}}$  assumes that the platformer's standardisation cycle  $\tau_{\text{ext}}$  is observable to the firm exploiting the time differential. When  $\tau_{\text{ext}}$  becomes irregular or unpredictable, the Samsung-form rent extraction strategy becomes unstable, and  $V_3$  collapses regardless of the firm's manufacturing capability.
- $V_2 = k_2 \cdot G_2 \cdot T_2 \cdot (1 - F)$  assumes that fragmentation  $F$  can be expressed as a scalar friction. In practice, fragmentation operates through multiple independent mechanisms (version drift, region splits, OEM modifications), and the scalar approximation is a first-order simplification. The formula's robustness depends on these mechanisms operating multiplicatively rather than additively.

The four scope conditions are not exhaustive, but they identify *where* the formal propositions are most likely to fail. A platformer analysis that ignores governance-criticality weighting ( $G_2$ ), assumes successful isolation as automatic ( $T_3$ ), treats the external platformer's cycle as fixed ( $T_4$ ), or treats fragmentation as a single scalar ( $V_2$ ) will produce conclusions that the underlying structural propositions do not support. The propositions are offered for use under their scope conditions; outside those conditions, the analysis requires modification or supplementary instruments.

This treatment of the formulas as *falsifiable structural propositions* — rather than as either calibrated empirical models or as illustrative metaphors — is the framework's response to the question: what would constitute evidence against this framework? The answer is: empirical findings that any of the scope conditions above are violated while the formal propositions continue to be asserted. The framework is therefore committed to revising specific propositions when scope-condition violations are observed, while preserving the architectural commitment that the conversion of spatial structure ( $H_a$ ,  $H_b$ ,  $H_c$ ) into

temporal discipline (T) into value capture (H<sub>d</sub>) is the operational form of platformer sovereignty.

### 3.7 Falsification Anchors: When the Framework Fails

The scope conditions of §3.6 identify *where* propositions may fail. This section identifies *what failure looks like*. Each proposition in the toolkit carries an implicit negation — a condition under which the proposition's predicted outcome reverses or evaporates. Identifying these negations is the framework's commitment to falsifiability: the propositions are not merely consistent with observed outcomes, they predict outcomes that, if reversed, would falsify the propositions.

*G<sub>1</sub> falsification anchor.* If a platformer increases vertical-stack discretion ( $\alpha_j$ ) without sustaining the temporal pulse (T<sub>1</sub>), the formula predicts that revenue (V<sub>1</sub>) collapses. The mechanism: vertical penetration without periodic ecosystem reset accumulates platform-specific debt without forcing market renewal, and the height-and-pulse capture configuration breaks down. The framework therefore predicts that a vertically-penetrated platformer that abandons annual or near-annual update discipline cannot sustain V<sub>1</sub>-form revenue.

*T<sub>1</sub> falsification anchor.* If a platformer increases update frequency (raises  $\omega_{reg}$ ) without proportionate increases in pulse amplitude (A<sub>amp</sub>), the formula predicts that ecosystem participants stop synchronising to the platformer's clock. The mechanism: high-frequency low-amplitude updates produce noise rather than signal; developers cannot prioritise rapid succession of minor changes, and the relational time (§3.3) that the platformer constitutes through its pulse becomes unreadable to the ecosystem. The framework predicts that excessive update frequency degrades, rather than improves, the platformer's temporal authority.

*T<sub>3</sub> falsification anchor.* If consortium governance is attempted at high structural coupling (A<sub>4</sub> large) without H<sub>a</sub> + H<sub>c</sub> discipline holding p high, the formula predicts catastrophic integration failure. This is the structural form of the Symbian outcome:  $T_3 = p^{A_4}$  falls toward zero as p drops below threshold. The framework predicts that any future mobility consortium operating at the scale of AUTOSAR, Catena-X, or Uranos *without machine-readable conformance enforcement* will reproduce the Symbian failure mode regardless of member firms' individual capabilities.

*T<sub>4</sub> falsification anchor.* If the external platformer compresses its standardisation cycle ( $\tau_{ext}$  shrinks toward  $\tau_{self}$ ), the time-difference rent ( $\Delta\tau$ ) approaches zero, and V<sub>3</sub> collapses. The framework predicts that a manufacturing platformer whose competitive advantage depends on preemption velocity (the Samsung-form) cannot sustain that advantage against a platformer that has adopted continuous-deployment discipline. This is a structurally testable prediction: as Google compresses Android standardisation cycles through AAOS continuous-update mechanisms, Samsung-form rent extraction in adjacent industries should observably decline.

*H\_d falsification anchor.* If a platformer captures value at  $H_d$  without operating  $H_a$ ,  $H_b$ ,  $H_c$  (the time-control handles), the formula predicts that capture is unstable. The mechanism:  $H_d$  harvests value from time produced by the other three handles; without the upstream handles, no time is produced to harvest. The framework predicts that pure value-capture platforms (advertising aggregators, payment processors) cannot sustain platformer sovereignty unless they acquire — or coordinate access to — the time-control handles upstream of their capture point.

These five anchors define the empirical commitments of the toolkit. The framework is not merely descriptive (G, T, H, V observed in cases); it is *predictive* (specific failures expected when paired conditions are violated). The mobile-industry history (Apple, Google, Samsung successes; Symbian failure) is consistent with these predictions. The mobility-industry transition will provide the next test.

## 4. The CPS Value Emergence Matrix and the Three-Loop Model

### 4.1 The CPS Value Emergence Matrix

The analytical toolkit of Section 3 diagnoses how platformers exercise sovereignty. The framework's prescriptive contribution requires a second analytical move: the reorganisation of platform governance around *two independent boundaries*.

The CPS Value Emergence Matrix (Cyber-Physical Symbiosis) is constructed from two axes:

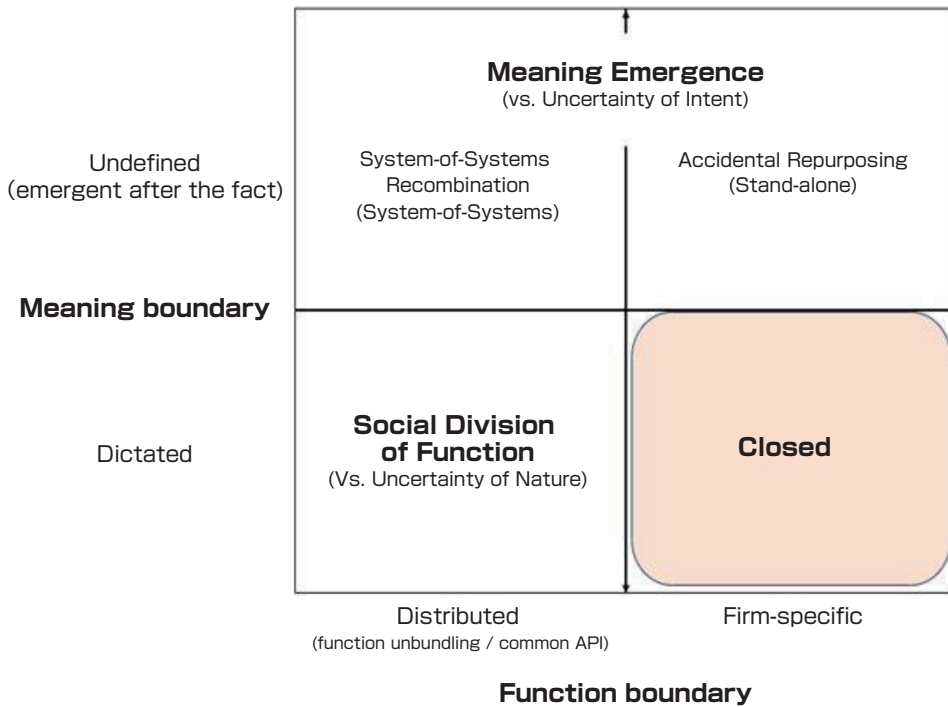
- *Horizontal axis:* the boundary of functional division of labour (closed / distributed)
- *Vertical axis:* the boundary of meaning-definition (dictated / undefined)

These axes are *not* the same as the typology's two axes (Section 2). The typology distinguishes firms by OS holding and integration patterns. The matrix distinguishes platforms by the *boundaries* they draw around function and meaning. A single firm can occupy different positions in the matrix at different layers of operation.

The two axes generate four quadrants:

- **Q1 (Accidental Repurposing):** function closed, meaning undefined. The Ford Model T as canonical example: a vehicle whose function was determined by Ford's vertical integration but whose meaning (rural use, agricultural conversion, mass-motorisation lifestyle) was generated by users without Ford's intention.
- **Q2 (Systematic Recombination — “The Goal”):** function distributed, meaning undefined. The framework's prescriptive destination. Multi-stakeholder coordination — industry, government, academia, citizens (*sankan-gaku-shimin*) — generates ecosystem-scale recombination of meaning on a shared functional infrastructure.
- **Q3 (Immunity Trap):** function distributed, meaning dictated. Google's AAOS configuration. Open APIs that extend reach, but meaning constrained by platformer interface specifications. The trap is that openness without responsibility cannot produce the legitimacy required for physical-safety industries.

- **Q4 (Enclosure Trap):** function closed, meaning dictated. Tesla, Apple. Strong vertical integration with strong intent over user experience. The trap closes when external value-creation opportunities exceed the capacity of any single firm to capture.



**Figure 5.** The CPS Value Emergence Matrix: Two boundaries — function (horizontal) and meaning (vertical) — define four quadrants

Source: author.

#### 4.2 The Q1 → Q2 Horizontal Connection

The matrix's most original analytical contribution is the *Q1 → Q2 connection*. Accidental repurposing in Q1 — Ford Model T's agricultural conversion, Honda Soichiro's post-war motorbike origin — demonstrates that users will generate unanticipated meaning on platforms whose function is held closed. The transition from Q1 to Q2 occurs through *visualisation* — the rendering of user-generated meaning into a form that multi-stakeholder coordination can address. Aisin's *Choisoko* (demand-responsive transit for ageing rural communities, where social-participation rates increased and medical-cost trajectories were measurably altered) is offered as a contemporary case where the unanticipated value was visualised and entered into systematic recombination by industry-government-academic-citizen coordination.

The connection is not the inevitable result of accidental innovation. It is the institutional capacity to *visualise* user-generated value and to convert that visualisation into coordinative recombination. The capacity for visualisation is itself a governance achievement.

### 4.3 Theoretical Inheritance: Two Independent Uncertainties

The matrix's two axes correspond to two analytical dimensions that strategic-management research has developed independently but never integrated. The vertical axis corresponds to what Numagami (2004) and Takeishi, Aoshima, & Karube (2012) call the *uncertainty of intention* — the uncertainty about how use, social value, and ecosystem meaning will emerge through inter-subjective interaction. The horizontal axis corresponds to the *uncertainty of nature* — the uncertainty of technical capability reduced through research, development, and engineering practice.

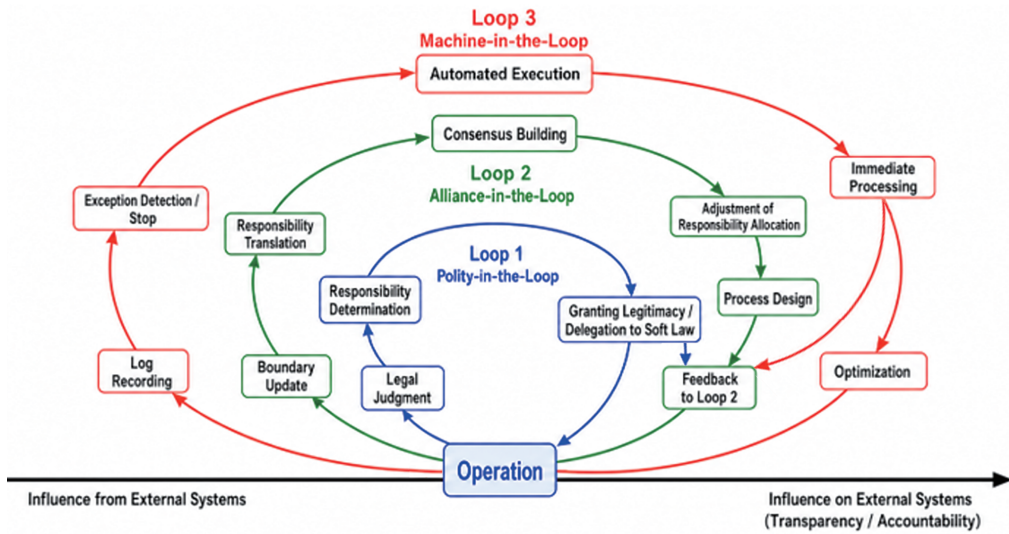
Existing theory has treated these uncertainties as features of a single, firm-level innovation process. The sense-making tradition (Weick, 1995) and the effectuation framework (Sarasvathy, 2001) have developed accounts of meaning generation under uncertainty. The product-architecture tradition (Ulrich, 1995; Fujimoto, 1997; Henderson & Clark, 1990) and the platform-strategy tradition (Gawer & Cusumano, 2002; Parker, Van Alstyne, & Choudary, 2016; Cusumano, Gawer, & Yoffie, 2019) have developed accounts of technical division of labour. The traditions remain *separate* because both processes have been theorised within a single firm's boundary. The CPS Value Emergence Matrix proposes their integration through the recognition that the two processes operate across *different boundaries*: meaning across the firm-society boundary, function across the firm-network boundary.

### 4.4 The Three-Loop Model

The matrix's Q2 cannot be occupied by any single firm. It requires institutional architecture — the capacity to coordinate multiple stakeholders across multiple time scales while preserving the responsibility framework that physical-safety industries demand.

The framework proposes the *Three-Loop Model* as the institutional form for this coordination. Three concentric loops, each operating at a different time scale and a different level of responsibility:

- **Inner loop (Polity-in-the-Loop)** — Centre. The core of legal responsibility: human life, criminal law, strict property rights, fundamental social responsibilities. Largest mass; slowest rate of change. Failure handling: court decision, institutional reform.
- **Middle loop (Alliance-in-the-Loop)** — The sovereign coordinator. Organisational alliance of industry, government, academia, and citizens. Operates as buffer: it translates the inner loop's requirements into operational thresholds the outer loop can execute, and it adjudicates failures that exceed the outer loop's containment capacity. Failure handling: re-contracting, re-design.
- **Outer loop (Machine-in-the-Loop)** — High-speed execution. Algorithms, code, data circulating at machine speed; immediate execution under pre-existing rules. Bears no result-responsibility. Failure handling: exception detection, system halt, log recording.



**Figure 6.** *The Three-Loop Model of CPS Governance: Concentric loops integrating speed, coordination, and legal accountability<sup>2)</sup>*  
Source: author.

The Three-Loop Model differs from existing models in a specific way. The agile-governance model proposed by the Japanese Ministry of Economy, Trade and Industry (METI, 2022) operates two concentric loops at homogeneous dimensions: rules-revision and operations-improvement. The DEOS Loop (Tokoro, 2010) operates two loops within the dependability-engineering framework: normal operation and incident response. Both are framed within a single dimension. The Three-Loop Model is distinctive in coordinating across *heterogeneous* dimensions: the absolute legal-responsibility constraint of the inner loop is qualitatively different from the high-speed algorithmic activity of the outer loop, and the Alliance-in-the-Loop is the only layer that can persistently translate between heterogeneous dimensions.

#### *Failure Modes of the Three-Loop Model*

The Three-Loop Model is not self-implementing. The framework's prescriptive proposal is therefore not complete without explicit registration of how the institutional architecture can fail. Three failure modes deserve recognition, each linked to a specific governance handle through which it can be countered.

*Failure Mode 1: Alliance fragmentation.* The middle loop's coordinative capacity depends on the willingness of member firms to suppress their own design preferences in favour of the common rule (the  $\gamma_i$  variable from  $G_3$  in Section 3). When  $\gamma_i$  declines — when individual firms prioritise differentiating features over consortium discipline — the alliance fragments and the failure structure of Symbian (Section 3) recurs at industry scale. The countermeasure operates through  $H_a$  (conformance gate quality): a strong, machine-readable, transparent conformance regime makes deviation visible and costly, preserving  $\gamma_i$  even when individual incentives diverge.

*Failure Mode 2: Alliance capture.* The middle loop can be co-opted by the dominant

platformer — through funding asymmetry, technical-expertise asymmetry, or sheer scale of participation. Captured alliance returns the configuration to Q3 (Immunity Trap): the appearance of consortium governance with the substance of single-firm rule. The countermeasure operates through H\_b (distribution diversity) and H\_c (runtime isolation): distribution channels and runtime boundaries that prevent any single member from concentrating operational authority, even when nominal alliance structure is preserved.

*Failure Mode 3: Polity inertia.* The inner loop's slow rate of change is its strength — legal responsibility, criminal accountability, fundamental rights cannot be revised at machine tempo. But the same slowness, when excessive, can suffocate the outer loop's adaptive capacity, producing institutional sclerosis: an industry that cannot evolve because its inner-loop constraints have not been updated to permit evolution. The countermeasure operates through H\_d (transparent capture metrics): visible, auditable evidence of where inner-loop constraints are binding outer-loop adaptation, providing the empirical basis on which the inner loop can be selectively updated through the middle loop's coordinative process.

The three failure modes and their countermeasures demonstrate that the institutional architecture and the analytical toolkit (G, T, H, V) are mutually constitutive. The toolkit identifies the variables through which the architecture can succeed or fail; the architecture identifies the institutional form within which the toolkit's variables can be operated at scale. The framework's prescriptive proposal is therefore not a single design but a *design space* — a set of configurations within which the variables can be tuned to counter the failure modes that local conditions render most threatening.

#### 4.5 Alliance-in-the-Loop, Not Human-in-the-Loop

I wish to distinguish *Alliance-in-the-Loop* from the more familiar *Human-in-the-Loop* used in AI ethics and machine-learning literature. Human-in-the-Loop assumes a single human (or a small set of humans) making real-time judgement interventions on AI outputs. Alliance-in-the-Loop assumes a multi-organisational consortium making coordinative decisions through agreement, contract, and process design.

The distinction matters. AI ethics frameworks that locate accountability in human-in-the-loop oversight cannot scale to the System-of-Systems governance that mobility requires. The volume of decisions, the heterogeneity of affected populations, the temporal distribution of consequences — these features exceed what any single human can handle. The Alliance-in-the-Loop is the framework's response: a coordinative architecture operating on agreement and process, rather than real-time judgement, scaling to industry-level governance.

#### 4.6 Automation versus *Jidoka* (Autonomation)

A specific distinction within the Three-Loop Model deserves explicit registration. The Toyota Production System tradition (Ohno, 1978) has long distinguished between *automation* (the replacement of human work by machine work) and *jidoka* (often translated

as *autonomation* — the construction of systems in which machines stop when they detect abnormality, preserving human judgement in the exception layer).

The framework argues that the Three-Loop Model's outer loop is *jidoka*, not automation. The machines of the outer loop operate at machine speed under pre-existing rules. *They do not replace human judgement; they preserve human judgement for the cases that exceed pre-existing rules.* When the outer loop encounters an exception, it halts and surfaces the exception to the middle loop, where the Alliance-in-the-Loop addresses it through coordinative process. The platformer-governance regime that conflates automation with *jidoka* — treating algorithmic decision-making as fully autonomous — has mis-engineered the relationship between the layers.

*Observational test for jidoka governance.* The framework's distinction between automation and *jidoka* becomes operationally testable through the exception-handling behaviour of the system. A platformer that operates *jidoka* can be observed to: (i) halt affected subsystems within bounded detection-to-halt times when anomalies are detected (not continue execution under degraded conditions); (ii) escalate exception events to a coordinative body (not absorb them within the platformer's unilateral discretion); and (iii) record exceptions with tamper-resistant evidence accessible to the middle loop (not retain them in private logs). A platformer that fails any of these three behaviours is operating automation, not *jidoka*, regardless of how the configuration is described. The framework predicts that platformers operating consumer-information automation (Google's earlier configuration, Meta's content moderation) cannot directly transfer to mobility-industry *jidoka* without architectural modification that satisfies these three observational tests.

## 5. The Receptive Platform and *Juseki* Governance

### 5.1 Governance Without Responsibility

The framework's prescriptive synthesis begins from a diagnostic recognition. Type 2 platformers — Google in particular — operate a configuration termed *governance without responsibility*: the extraction of value from platform operation while disclaiming the physical-liability consequences. The architectural form of this configuration is the spatial separation produced by the Vehicle Hardware Abstraction Layer (placing physical responsibility outside the platformer's domain) combined with the temporal separation produced by Project Treble (decoupling the platformer's update clock from the OEM's safety-validated cycle). Together, they produce a configuration in which the platformer monetises the fast clock while the OEM carries the obligations of the slow clock.

This configuration is structurally enabled by the immunity regime that Section 230 of the U.S. Communications Decency Act (CDA 230) established for consumer-information platforms in 1996. The judicial trajectory that includes *Lemmon v. Snap Inc.* (2021) — in which platform design was held to be product, not protected expression — and the EU New Product Liability Directive (2024) — which defines software, operating systems, and AI

components as products subject to product-liability law — together demonstrate that this immunity regime cannot extend to the mobility industry without structural transformation. The Japanese legal scholar Narihara (2019, 2023) has articulated this transformation through the concept of the *right to hack* — the OEM’s right to refuse and legally contest platformer-imposed specifications, on the grounds that the OEM bears the physical responsibility that the platformer would otherwise externalise.

## 5.2 The Receptive Platform

The framework’s prescriptive concept is the *receptive platform* — a platform architecture whose constitutive principle is *juseki* (受責), the active taking-on of responsibility, operating through the governance principle of *ANSHIN* (安心).

The receptive platform is the institutional answer to the structural problem of governance without responsibility. Where Type 2 platformers operate governance without responsibility, the receptive platform operates *governance through responsibility*. The shift is not rhetorical. It is structural and operational. The receptive platform:

- accepts responsibility for the physical, legal, and social consequences of platform operation;
- internalises this responsibility into the technical architecture through five functional clusters: evidence (SBOM, telemetry, logs), proof (machine-readable compliance), key (PKI and Root-of-Trust), execution (isolation and rollback APIs), and time (industry-wide tempo coordination);
- operates this responsibility through the Three-Loop Model’s coordinative architecture; and
- recovers the cost of operating responsibility through what the framework calls a *digital social tax* — the redefinition of platform fees as a contribution to responsibility-infrastructure maintenance, rather than rent on a monopolistic position.

The shift from “irresponsibility” to *juseki* is the framework’s most demanding prescriptive proposal. It requires a fundamental reorientation of what platformers do and how they are governed. The framework does not pretend this reorientation is easy or that any single firm can achieve it alone. It argues that the institutional architecture for the reorientation — the consortium operating the five functional clusters within the Three-Loop Model — is the structural form through which the reorientation becomes operable.

## 5.3 Undefined-by-Design

The Q2 architecture requires not only the institutional Three-Loop Model but also an *architectural posture* — a deliberate design decision to leave the meaning-definition boundary undefined. The framework calls this posture *Undefined-by-Design*.

Undefined-by-Design is the deliberate refusal to pre-specify the use, purpose, and meaning of platform components. The Q4 firms define their meaning fully; this is the source of their competitive strength in single-firm terms and the source of their structural limitation in

ecosystem terms. The Q2 architecture *refuses* to define meaning at the platform level, on the grounds that the value of the platform grows when its meaning is generated by users and coordinated by alliance.

This is not the same as leaving the platform passive. The Q2 architecture *actively designs* the functional boundary while *actively refraining* from defining the meaning boundary. The function is rigorously specified — APIs, safety properties, certification regimes, evidence standards. The meaning is rigorously left open — for users, for partners, for unanticipated alliances, for stakeholder coordination through the middle loop.

#### 5.4 Full Self-Driving as a Strong Promotive Condition for Q2

A claim that the framework would advance, with appropriate qualifications, concerns full self-driving (FSD). The conventional industry framing treats FSD as a technical objective — the result of progressive improvement in AI, sensors, and computation, to be achieved when the technology matures. The framework argues that this framing under-determines the causal role of FSD in the platform-economic transition. FSD is not merely a technical outcome to be awaited; it functions as a *strong promotive condition* for the separation of function and meaning at the platform level — though not, as I will clarify below, as a strict necessary or sufficient condition.

The structural argument has two stages.

*First stage: human operation as obstacle to function-meaning separation.* As long as a human driver is actively engaged in operational control, the *function* of the platform (movement A to B) is bound to the *meaning* (the purpose for which movement is undertaken). The driver's act of operating the vehicle is, simultaneously, the act of giving the vehicle its purpose. This binding is structural, not merely descriptive: the operating subject and the meaning-generating subject coincide. The function-meaning boundary that Q2 requires cannot be drawn while the driver remains the operational subject. *Q2 is therefore structurally difficult to occupy while human operation predominates*, regardless of how sophisticated the platform's HMI design or how flexible its UX architecture.

This first-stage claim faces an obvious objection. Partial function-meaning separation is achievable through HMI design, regulatory architecture, and UX governance even with human operation. The framework concedes this. The claim is not that no separation is possible without FSD, but that the separation is *partial, contingent on individual driver behaviour, and structurally bounded* by the human-operation constraint. Q1 (Accidental Repurposing, where Ford Model T users generated meaning their function had not anticipated) is precisely the form that limited function-meaning separation takes under human operation. Q2 — the *systematic* coordination of distributed function with multi-stakeholder meaning recombination — requires a stronger separation than human-operation regimes can structurally provide.

*Second stage: FSD as enabler, not sufficient cause.* Full self-driving, by transferring operational control from driver to system, makes the function-meaning separation

architecturally possible at full scale. The operating subject (the system) becomes distinct from the meaning-generating subject (the passenger), and the system's responsibility framework becomes structurally tractable. However — and this qualification is critical — *FSD does not, by itself, produce Q2*. An FSD platform can equally produce Q4 (Enclosure Trap) if its meaning-definition boundary is closed. Tesla's Robotaxi configuration, in which the platform operates without driver but the entire meaning of mobility is dictated by Tesla's service design, is precisely a Q4 outcome reached through FSD. The architectural posture of Undefined-by-Design (§5.3) is what distinguishes Q2 from Q4 *after* FSD has enabled the separation. The two conditions — FSD and Undefined-by-Design — are jointly required for Q2; neither alone is sufficient.

The two-stage argument supports a careful claim: FSD is a *strong promotive condition* for Q2 (substantially lowering the structural obstacles to function-meaning separation) but is not a necessary condition (limited separation is possible without it) and is not a sufficient condition (without Undefined-by-Design, FSD produces Q4 rather than Q2). The framework's prescriptive interest in FSD therefore concerns *not whether the technology will be achieved* but *whether its achievement will be paired with the architectural posture that converts the technological capability into the institutional outcome the framework proposes*. A subsidiary structural observation: the FSD-compatible architectural substrate is the battery-electric vehicle (BEV). BEVs operate on continuous, immediately reversible motor torque control — the physical substrate that supports continuous, non-deterministic software-based control. Internal-combustion-engine (ICE) and hybrid-electric (HEV) vehicles operate on discrete, irreversible combustion events coupled with gear-based transmissions — substrates that resist the software-based continuous control that FSD requires. While Multi-Pathway industrial policy (MPW) retains ICE/HEV/PHEV/FCEV alongside BEV as parallel technology paths, the structural conditions for FSD are dispersed across heterogeneous substrates, and the platformer transition that FSD would promote is structurally postponed.

## 6. Conclusion: Implications for Digital Colonisation

This paper has developed a theoretical framework for analysing platformer governance in the mobility industry. The framework's contribution is structured across three levels.

At the analytical level, the toolkit of G, T, H, and V provides a diagnostic vocabulary through which any platformer's sovereignty regime can be examined. The master expression  $\text{Profit} = f(\text{What, When, How}) \rightarrow \text{Where}$  compresses the structural form by which platform value is captured.

At the structural level, the CPS Value Emergence Matrix reorganises platform governance around two independent boundaries — function and meaning — and identifies the Q1  $\rightarrow$  Q2 connection as the empirically observable mechanism by which user-generated value enters coordinative recombination.

At the prescriptive level, the Three-Loop Model gives institutional form to multi-stakeholder coordination, and the receptive-platform concept provides the architectural target toward which the institutional construction would point.

The framework's location relative to the digital-colonisation literature merits explicit registration. Ozalp et al. (2022) have documented, with empirical precision, the four-stage process by which Big Tech platforms enter highly regulated industries — from data-infrastructure provision through data capture and insight generation to the commercialisation of new products and services. The mobility-industry case is, in that framework, one further empirical site of encroachment — distinguished from the healthcare and education cases by the addition of physical-safety and product-liability constraints, and by the specific combination of platform-economic logic with physical-engineering responsibility.

What this framework attempts to add is twofold. First, a temporal architecture — the T<sub>1</sub>–T<sub>4</sub> disciplines, the time-political infrastructure, the five functional clusters — through which platformer-governance dynamics can be diagnosed and re-engineered. Second, a constructive proposal — the receptive platform, the Three-Loop Model, the consortium operating computational jurisprudence — for the institutional form through which platformer encroachment can be checked without forfeiting the speed and scale that platform-economic logic produces.

Neither contribution is final. Both are offered as instruments through which the digital-colonisation question can be re-examined with greater structural precision than the healthcare and education cases alone permitted.

This paper ends with a question that re-orientates the entire framework. The conventional industry question — "when will full self-driving be achieved?" — is the wrong question. The right question is structurally different: Who will control the receptive-platform infrastructure that operates the consortium's five functional clusters within the Three-Loop Model, and who will design the social-impact circuit through which the systematic recombination of Q2 returns value to the infrastructure?

This is the question whose answer will define the next phase of the mobility industry's sovereignty regime. The framework developed in this paper offers no prediction of the answer. It offers the analytical instruments through which the question can be posed at the level of structural precision that the mobility industry's transition requires.

## Appendix

- 1) "OS governance holding" means the durable grasp of conditional-setting rights — over CTS, APIs, update discipline, distribution channels, certification — not legal ownership of kernel code. Google holds Android's governance without owning its kernel.
- 2) The Three-Loop Model differs from existing models (METI agile governance, DEOS Loop) in coordinating across heterogeneous dimensions: the absolute legal-responsibility constraint of the inner loop is qualitatively different from the high-speed algorithmic activity of the outer loop, and the Alliance-in-the-Loop is the only layer that can persistently translate between them.

## References

- Barney, J. B. (1991). Firm resources and sustained competitive advantage. *Journal of Management*, 17(1), 99-120.
- Cusumano, M. A., Gawer, A., & Yoffie, D. B. (2019). *The business of platforms: Strategy in the age of digital competition, innovation, and power*. Harper Business.
- Eisenhardt, K. M. (1989). Making fast strategic decisions in high-velocity environments. *Academy of Management Journal*, 32(3), 543-576.
- European Union. (2024). Directive (EU) 2024/2853 on liability for defective products [New Product Liability Directive].
- Fujimoto, T. (1997). *Seisan shisutemu no shinkaron: Toyota jidōsha ni miru soshiki nōryoku to sōhatsu purosesu* [The evolution of production systems: Organizational capability and emergent processes at Toyota]. Yuhikaku.
- Gawer, A., & Cusumano, M. A. (2002). *Platform leadership: How Intel, Microsoft, and Cisco drive industry innovation*. Harvard Business School Press.
- 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.
- Lemmon v. Snap Inc., 995 F.3d 1085 (9th Cir. 2021).
- Lieberman, M. B., & Montgomery, D. B. (1988). First-mover advantages. *Strategic Management Journal*, 9(S1), 41-58.
- Ministry of Economy, Trade and Industry, Japan. (2022). *GOVERNANCE INNOVATION Vol. 3: Agile governance no gaiyō to genjō* [Overview and current state of agile governance]. Tokyo: Ministry of Economy, Trade and Industry.
- Narihara, K. (2019). API to hō: Purattofōmu jidai no kihan keisei [APIs and the law: Norm formation in the platform era]. *Hōritsu Jihō*, 91(10), 12-18.
- Narihara, K. (2023). Haku suru kenri: Gijutsu hyōjun e no hōteki taikō kanōsei [The right to hack: Legal contestation of technical standards]. *Hōgaku Kyōshitsu*, 514, 44-51.
- Numagami, T. (2004). *Senryaku kōchiku to iu jissen* [The practice of strategy construction]. Hakutō Shobō.
- Ohno, T. (1978). *Toyota seisan hōshiki* [Toyota production system]. Diamond.
- Ozalp, H., Ozcan, P., Dinckol, D., Zachariadis, M., and Gawer, A. (2022) “Digital Colonization” of Highly Regulated Industries: An Analysis of Big Tech Platforms’ Entry into Health Care and Education. *California Management Review*, 64(4), 78-107.
- Parker, G. G., Van Alstyne, M. W., & Choudary, S. P. (2016). *Platform revolution: How networked markets are transforming the economy*. W. W. Norton.
- Sarasvathy, S. D. (2001). Causation and effectuation: Toward a theoretical shift from economic inevitability to entrepreneurial contingency. *Academy of Management Review*, 26(2), 243-263.
- Takeishi, A., Aoshima, Y., & Karube, M. (2012). *Innobēshon no riron to keiei* [Theory and management of innovation]. Yuhikaku.
- Teece, D. J. (2007). Explicating dynamic capabilities: The nature and microfoundations of (sustainable) enterprise performance. *Strategic Management Journal*, 28(13), 1319-1350.
- Tokoro, M. (2010). *Open systems dependability: Challenges and possibilities*. JST CREST DEOS Project Report, Tokyo: Japan Science and Technology Agency.
- Ulrich, K. T. (1995). The role of product architecture in the manufacturing firm. *Research Policy*, 24(3), 419-440.
- Weick, K. E. (1995). *Sensemaking in organizations*. Thousand Oaks, CA: Sage Publications.