

AI-Economy-Architecture-V3

Separate commerce infrastructures sharing coordinated resource capacity

Ashish Warudkar | GUDIYA Cognitome | Version 3 | September 2026

The March 2025 starting point

In March 2025, I articulated an intuition that I called 'AI-Vertigo' and described it in a video. Different functions within an enterprise would acquire AI capabilities at different rates. Purchasing might accelerate at one cadence, marketing at another, and human resources or operations at a third. Their responsibilities would remain interconnected even as their ability to generate and respond to work changed unevenly.

The concern was organizational disorientation: one function could create demands, expectations, or commitments that another function could no longer absorb at the same pace. Improving each function independently would not necessarily improve the enterprise as a whole. The relationships between functions would need renewed attention.

Version 3 extends that hypothesis to enterprises serving both conventional digital commerce and agentic commerce. Their interaction workloads may justify separate digital infrastructures, yet both channels draw on shared inventory, production, logistics, and other resources. GUDIYA proposes coordinating their combined commitments so that infrastructure separation protects service quality without allowing either channel to overpromise common capacity.

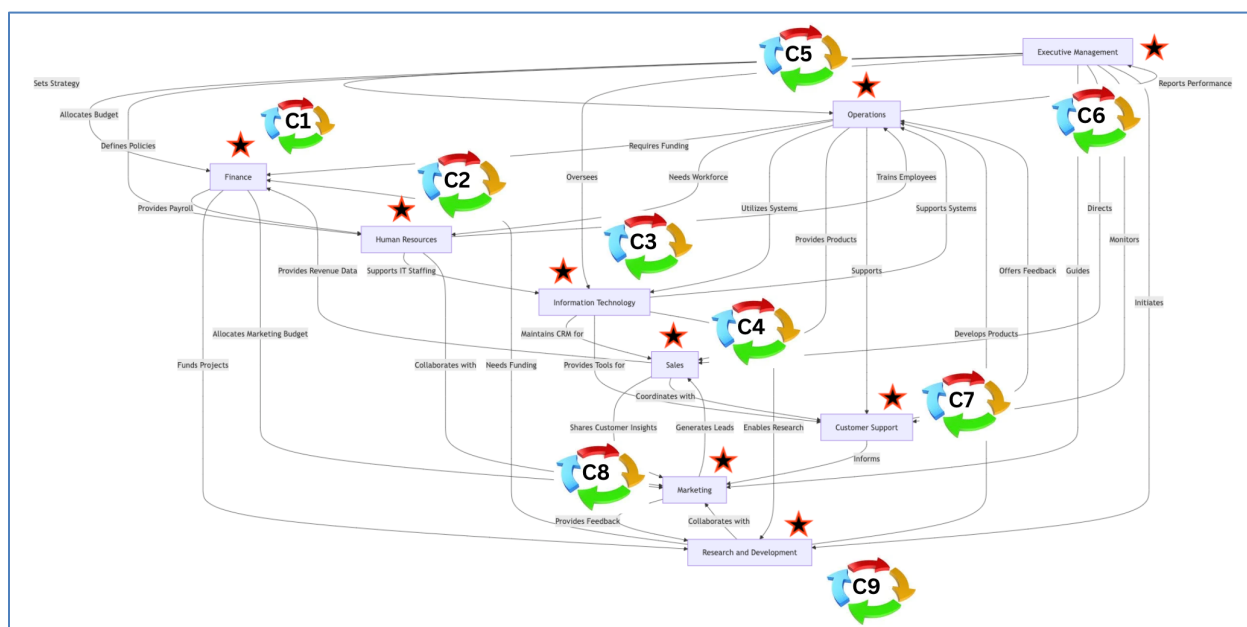


Figure 1 : Ai-Vertigo happens when different parts of the Enterprise accelerate at differential cadence

A more precise statement of the hypothesis

AI-Vertigo arises when the acceleration of coupled activities outpaces the system's ability to coordinate their commitments, feedback, and fulfillment. Different cadences alone are insufficient to produce instability. The vulnerability appears when interactions exceed available margins or defeat the arrangements that reconcile those cadences.

At enterprise scale, the mismatch can occur between departments. At ecosystem scale, it can occur between agents, platforms, suppliers, and shared resource networks. The mechanism under examination is the same; the boundaries and dependencies extend further.

The architectural principle is simple: separate channels where their workload and service requirements justify it, while coordinating the obligations they create against shared resources. The following design is a proposal, building on established infrastructure isolation and transaction practices.

Three layers of an interacting economy

The three-layer view distinguishes agentic activity, digital services, and resource fulfillment. We introduce two commerce channels within this structure. Human-led commerce enters through its digital interfaces; autonomous agents enter through an agentic interface. Their supporting service infrastructures may be isolated, while both connect to the resource channel below. Channels are alternative paths through the architecture, not additional vertical layers.

Layer	Function	Characteristic constraint
1 Agentic activity	Interpret intent, plan, delegate, adapt, and initiate work.	Decision latency, concurrency, permissions, and execution budgets.
2 Digital services	Support separate digital-commerce and agentic-commerce channels; transact and record obligations.	Per-channel service capacity plus coordinated reservation authority.
3 Resources and fulfillment	Supply inventory, production, transport, labor, energy, and other deliverables.	Stock, physical throughput, schedules, recovery time, and availability.

Here, digital commerce means the conventional customer-facing channel, including established integrations; agentic commerce means the channel for delegated autonomous activity. Both are digital. A task may move between them as a person delegates to an agent or resumes control. Channel classification should follow the actual interaction and authority, not a permanently assigned customer label.



Figure 1 — Three layers with two service channels. Show people entering the digital-commerce channel and agents entering the agentic-commerce channel. Both lead through coordinated commitment accounting to a shared resource network. The Grid spans all three layers; it is not another mandatory processing hop.

Characteristic speed regimes

The agentic layer can generate work rapidly and in parallel; digital services process that work at bounded rates; fulfillment often takes longer. These are characteristic regimes, not a universal ranking. An API can process requests faster than a model can reason, and a physical control loop can operate faster than a business application. Every relevant operation needs its own timing and capacity description.

Activity rates also need meaningful units. Decisions per minute, requests per second, orders per hour, and deliveries per day cannot simply be subtracted. The Grid needs estimates of how upstream activity translates into downstream demand. One agent decision may generate hundreds of queries but no purchase; one accepted order may consume several scarce resources.

Two digital infrastructures when workloads justify them

An enterprise may choose one infrastructure for its conventional digital-commerce channel and another for agentic commerce. The purpose is to give each a suitable operating envelope. An agent's parallel searches, repeated evaluations, or bursts of delegated work should not automatically consume the capacity reserved for interactive customer service.

The case for separation rests on measured workload differences and service guarantees. Existing APIs, gateways, firewalls, and databases are not inherently limited to a human cadence. They are designed and configured around assumptions about traffic, concurrency, query cost, and transaction behavior. Agentic activity can invalidate those assumptions even when purchases do not increase proportionately.

Concern	Digital-commerce channel	Agentic-commerce channel
Interfaces	Web or mobile sessions and established integrations.	Explicit agent interfaces, delegated operations, and event delivery.
Identity	Customer, session, and transaction authority.	Customer plus agent identity, delegation scope, and execution limits.
Service envelope	Interactive latency and protected customer-facing capacity.	Query budgets, concurrency limits, burst allowances, and bounded outstanding work.
Processing	Capacity allocated to conventional workflows.	Independently scalable workers, queues, and query capacity where needed.
Transaction controls	Verified acceptance and clear customer status.	The same acceptance discipline, with explicit retry and asynchronous outcome contracts.

Separation can begin with distinct gateways, queues, worker pools, and quotas on shared infrastructure. Stronger isolation may add independently deployed services, caches, database read replicas, or networks. The enterprise should isolate the actual bottleneck: separate front doors accomplish little if both flood the same unprotected database connection pool.

This has a precedent in the bulkhead pattern: isolate resource pools so a fault or overload in one area is less able to impair others. GUDIYA applies that design reasoning to commerce channels while adding attention to their shared downstream obligations. Separation itself is not a new invention. [4]

Physical duplication is a choice, not a universal requirement. It adds cost and operational complexity, and can strand unused capacity. Shared components can remain where enforced budgets and failure boundaries provide sufficient protection. Conversely, service capacity reserved for one channel must remain protected when the other becomes overloaded.

Two entrances to a shared resource channel

Both commerce channels ultimately request capacity from the same enterprise or partner network. They may compete for stock, delivery slots, machines, loading bays, staff time, or digital fulfillment capacity. Their digital independence must therefore coexist with a coordinated account of commitments and reservations.

Shared resource channel does not mean one database, one owner, or one central server. It means that each resource and time window has an accountable authority and a consistent allocation rule. The enterprise may partition resources, distribute reservation services, or delegate bounded allowances. It must prevent the same capacity from being independently promised through both channels.

An illustration with 100 delivery slots

Suppose a retailer has 100 usable delivery slots tomorrow after existing obligations and a separately protected recovery reserve. In this simplified example, every order consumes one slot. The digital-commerce channel receives an allowance of 40, the agentic-commerce channel 40, and 20 remain unallocated for later assignment. Neither channel may advertise the whole 100 as its own remaining capacity.

After the digital channel accepts 30 orders and the agentic channel accepts 35, 65 slots are committed. The remaining capacity is 10 in the digital allowance, 5 in the agentic allowance, and 20 unallocated. If another 15 slots are assigned from that common pool to the agentic channel, its remaining allowance becomes 20 and the common pool becomes 5. Total remaining capacity is still 35.

Reallocation changes permission to promise; it does not manufacture capacity. Moving unused allowance between channels requires authoritative confirmation that the donor cannot keep spending it. During communication failures, a channel can continue within a safely delegated allowance, but cannot assume that extra common capacity is available.

Capacity must be counted in its proper dimensions

Real orders rarely consume identical resources. One delivery may need refrigeration, another special handling, and another a distant route. Admission must satisfy all relevant constraints and deadlines. A single order count can illustrate the principle but cannot replace resource-specific capacity accounting.

A practical invariant is that accepted commitments and valid outstanding reservation rights must fit within usable capacity for the relevant resource and window, with any protection margin accounted for once. Forecast demand informs allocation but does not become a reservation by itself. If usable capacity falls, the system must reassess existing commitments as well as stop unsupported new ones.

Allocation between channels is an explicit enterprise policy, potentially including guaranteed shares, priorities, and fair access. Machine-speed competition must not automatically give agentic requests all available inventory. Grid coordination enforces the authorized policy; it does not choose commercial priorities on the enterprise's behalf.

From separate access to coordinated settlement

Each channel should distinguish three events: receipt of a request, acceptance of a commitment, and resolution of the resulting obligation. A fast response can acknowledge receipt or propose a later slot without promising immediate fulfillment. This allows the agentic interface to remain responsive while the resource channel operates within its constraints.

The handoff contract

Before a transaction binds capacity, the request should establish who is acting, the authority delegated, the intended operation, its expiry, and the evidence required for acceptance and settlement. For a multi-agent workflow, the approved intent-group supplies the plan and limits; material changes in demand or dependencies require reassessment.

A person may begin a purchase on the website, delegate comparison to an agent, and finish checkout personally. The same business intent can therefore cross channels. A Gudiya-cookie links its lifecycle records, while operation identifiers distinguish legitimate new actions from retries. The effect identifier alone cannot make every call idempotent: a single effect may contain several different authorized purchases.

Retry protection must be enforced where the authoritative mutation occurs. A duplicated request arriving through the other channel must not create a second reservation merely because it entered through a different gateway. Established idempotent API techniques are useful here, but their identifiers, scope, retention, and semantics need explicit agreement. [5]

Separate reads and authoritative commitments

Agent queries may use caches or read replicas to reduce pressure, with freshness disclosed. But a displayed availability estimate is not a reservation. The commitment path must confirm capacity with its authority or

spend a valid delegated allowance. Otherwise stale views in two otherwise healthy infrastructures can create incompatible promises.

Multi-resource orders may need staged reservations and a recovery plan if one component fails. Capacity already consumed cannot simply be rolled back by changing a database record. The responsible participants must reconcile what happened and fulfill, cancel, compensate, or transfer obligations according to an accepted agreement.

What remains after interruption

Circuit breaking can stop further agent actions without erasing prior commitments. A reservation timeout also does not prove that an external action never occurred. Uncertain outcomes require reconciliation before capacity or payment obligations are released inappropriately. Each outstanding obligation needs an owner even if its agent, host, or model changes.

The shared resource channel should return settlement evidence to the appropriate owners and effect records, with only the information needed by each commerce channel. The two service infrastructures can remain operationally distinct while maintaining continuity of the customer's business outcome.

Why the interfaces matter

The highway analogy makes the relationship tangible. Traffic can occupy neighboring lanes at different speeds. A lane transition requires enough space and compatible movement in the receiving lane. An abrupt merge can force braking, which affects following vehicles and may disturb adjacent lanes. The local interaction creates consequences beyond the original participants.

In the AI economy, a handoff from either commerce channel must accommodate the receiver's current conditions. Authorization to access a service does not establish that a particular burst of work is supportable. An agent also differs from a vehicle: it can keep operating while producing thousands of additional requests. The interaction can multiply obligations rather than move one conserved unit between lanes.

The atmospheric analogy

Atmospheric layers moving at different velocities produce wind shear. Under suitable conditions, disturbances can develop into Kelvin–Helmholtz waves and turbulence. Their development depends on shear and the stabilizing conditions of the atmosphere. Differential velocity does not guarantee turbulence. National Weather Service descriptions provide the physical basis for this analogy. [1, 2]

We use cadence shear as a proposed GUDIYA term for pressure created when coupled activities generate downstream demands incompatible with the receiving system's cadence and capacity. This is an engineering analogy; atmospheric equations and physical energy measures cannot be transferred to cognition without a justified model.

From platform admission to ecosystem coordination

Amazon's reported restriction of Meta's Muse illustrates a platform enforcing its participation boundary. Its publicly reported position emphasized authorized, open participation and the provider's decision about access. This does not establish that Amazon acted because physical fulfillment was being overwhelmed. [3]

The Nadella excerpt supplied during this research raises a related question: what happens to existing service guarantees when agents generate traffic outside the assumptions of interfaces built for people? It points toward new access arrangements, infrastructure, and costs. That excerpt is contextual material; its wording is not reproduced here as an independently verified transcript.

GUDIYA extends the design question across participants. Even when every requester is authorized and every individual transaction is valid, their combined demands may exceed a shared service or resource. Admittance Control is the first boundary defense; ongoing coordination must also account for what admitted activity causes downstream.

The claim is therefore conditional: existing API infrastructure cannot automatically be assumed ready for a new agentic workload. The required response depends on the workload, its amplification, the service's operating envelope, and the obligations generated beyond it.

Following commitments through settlement

A Gudiya-Effect comprises an initiated action and its attributable consequences within a declared scope and observation horizon. Its lifecycle includes the obligations created, their owners, and evidence of resolution. The Gudiya-cookie connects those records across actants, enterprises, and layers. It is an identity and continuity mechanism, not by itself proof of causality or permission to act.

An effect is generally a graph. Work may branch, retry, merge with other workflows, or create subordinate obligations. Several effects may share a shipment or production run. Explicit couplings include calls, orders, reservations, and handoffs. Implicit couplings include competition for common capacity or shared state without direct communication between the competing agents.



Figure 2 —Trace a Gudiya-Effect from a human request through agent delegation, the agentic service channel, a shared reservation authority, and fulfillment. Preserve the same cookie across a later human-channel status check. Distinguish effect identity from individual operation identifiers.

Commitment and settlement are lifecycle responsibilities across the layers. An agent can propose an action without binding anyone. A transaction service can create the binding obligation. Resolution may then require evidence from a delivery, a payment service, and an acceptance process. A successful API response does not necessarily mean that the resulting obligation has been fulfilled.

A worked mismatch

Consider a simplified illustration in common units. Agents submit 1,000 distinct, valid one-unit orders per minute. A digital service can accept 500 per minute, while fulfillment can complete 100 per minute. If the service accepts at its maximum rate, unsettled obligations grow by 400 per minute, assuming no cancellations or capacity changes. If excess submissions are queued, the digital backlog also grows; if they are rejected, those rejections must not provoke uncontrolled retries.

Increasing digital throughput alone can worsen the fulfillment gap. A service can remain responsive while accepting promises the ecosystem cannot honor. Buffers may accommodate a temporary burst when spare future capacity exists, but a sustained demand excess eventually exhausts any finite buffer.

Settlement needs evidence and ownership

The Grid should distinguish proposed demand, accepted commitments, reservations, execution status, and verified settlement. A timeout means the outcome is uncertain until reconciled; it is not automatic proof of failure. Capacity should be released according to the relevant obligation's verified state, not held indiscriminately until every branch ends or released merely because a process stopped.

Latent consequences and incomplete visibility remain possible. Records must retain uncertainty, residual exposure, and later evidence without claiming to track every consequence forever. Actant migration or model replacement must preserve authorized identity, unfinished obligations, and relevant stability history.

How perturbations cross the layers

Suppose a delayed reservation response causes an agent to submit alternatives to several suppliers. Each supplier sees apparently legitimate demand. Inventory becomes reserved in several places while the original outcome remains unknown. Other agents detect reduced availability and respond by reserving still more alternatives. A local delay has now changed decisions across a resource neighborhood.

This example can contain reinforcing feedback: delay prompts additional activity, and that activity creates further delay. Other cascades can propagate through dependencies without a returning feedback loop. A missed component delivery can halt assembly and subsequently delay distribution. The Grid must distinguish the mechanism before choosing an intervention.



Figure 3 — Show an agent-channel burst contained by its service limits, alongside a resource shortage that affects both channels despite their isolation. Trace upward feedback and shared dependencies. Distinguish a bounded disturbance from an amplifying loop.

Endogenous and exogenous describe origin relative to a stated boundary. A storm may be external to an enterprise; its agents' synchronized retries are internal responses. Instability describes the resulting dynamics, such as growing deviations, persistent harmful oscillation, or departure from acceptable operating bounds. A disturbance does not inevitably produce instability.

Saturation has several meanings

Constraint	What the Grid must distinguish
Throughput	How quickly work can be processed or fulfilled.
Stock	Inventory or other consumables remaining and expected replenishment.
Concurrency	Simultaneously available slots, machines, connections, or bays.
Deadline capacity	Whether required work fits within the promised time window.
Recovery margin	Reserve available for faults, exceptions, and changing conditions.



Figure 4 — Show separate channel admission limits and a common resource constraint. Depict channel allowances, shared remaining capacity, and already accepted obligations. A breaker blocks selected new actions while settlement processing continues.

Capacity must be associated with a service or resource, location, time window, and uncertainty. Installed capacity is not the same as available capacity. Maintenance, reservations, product mix, and dependencies can change what is usable. A universal utilization percentage cannot represent all these constraints. Channel isolation can contain a request storm, but it cannot isolate both channels from every common warehouse, identity service, reservation authority, or other shared dependency.

Five stabilization disciplines across three layers

The Grid spans both commerce channels and their shared resources. Local controllers enforce channel budgets and resource-owner allowances; regional or broader coordination evaluates shared dependencies. Every transaction need not wait for a remote central controller. Common governance means coordinated authority, not mandatory central execution.

Discipline	Responsibility
Admittance Control	Apply channel-specific permissions and traffic budgets; admit commitments against coordinated resource allowances.
DDMC	Distributed Dynamic Margin Computation estimates each channel’s service margin and the combined fulfillment, deadline, and recovery margins.
DLPC	Distributed Dynamic Leverage Points Calculation compares intervention at a channel, shared service, reservation boundary, or resource.

Discipline	Responsibility
Cognitive Rx	Pace channel traffic, suppress duplicate operations, reconcile uncertain outcomes, or reallocate unused allowances within policy.
Circuit Breaking	Isolate selected channel actions or unsafe shared commitments; preserve settlement evidence and responsibility for accepted work.

Rendezvous Management at each boundary

An intent-group is a proposed execution plan; approval admits that plan under stated conditions. Before a consequential handoff, participants need compatible identities, authority, semantics, timing, and acceptance capacity. Pre-Rendezvous Conditioning prepares the interaction: reconcile stale state, confirm resource availability, bound concurrency, and establish how an uncertain outcome will be resolved.

Approval cannot be permanent immunity from changing conditions. Material plan changes, newly discovered dependencies, and capacity losses require reassessment. The Grid may defer new work, stage acceptance, or suspend selected actions within its delegated authority. Stopping a workflow does not remove commitments it has already made.

The operational loop

Sense activity and settlement evidence; update dependency and obligation records; forecast pressure across relevant horizons; compute margins; identify intervention points; apply authorized controls; and verify the result. Circuit breaking belongs explicitly in this loop as an escalation action when ordinary adjustments cannot contain the risk. Restoration requires reassessment rather than an automatic flood of queued work.

A dynamic capacity graph would associate resources with time-dependent capacity, reservations, forecast demand, and observed consumption. Linked effect graphs explain who generated obligations and who remains responsible. Forecasts are not reservations, and different participants must not independently spend the same capacity. Owners must supply authoritative state and timely updates; the Grid cannot infer every missing metric reliably.

Buffers and gradual enterprise participation

The earlier flywheel concept remains useful at the commitment boundary: buffer bursts of proposed work, suppress duplicates, pace release, and reconcile outstanding attempts. At digital services, bounded queues and concurrency controls perform related functions. These measures buy time only when their limits and exit conditions are explicit.

Buffers and flywheel services can be budgeted separately for each commerce channel. At fulfillment, their counterpart must be backed by real slack: inventory reserves, available production windows, spare transport, or agreed alternative capacity. A digital record cannot create a missing truck. Every buffer needs a maximum exposure, an acceptable delay, and an escalation rule before its margin is exhausted. Long-run admitted demand must remain compatible with sustainable fulfillment.

Participation can expand with operational value

Participation	Potential support
Digital actants and services	Identity continuity, observable dependency tracking, pacing, retry control, and transaction reconciliation.
Acceptance envelopes	Admission against owner-declared quantities and time windows without exposing detailed internal operations.
Selected resources	Forecasting and coordination around chosen bottlenecks and their reservations.
Broader resource visibility	More extensive fulfillment balancing and recovery within explicitly delegated authority.

An enterprise can begin by exposing both channel gateways, then add a resource acceptance envelope or selected bottlenecks. For example, it might disclose capacity for another 200 orders in a delivery window without revealing staffing or supplier details. That envelope needs scope, validity, confirmation, and revision rules. It is a consumable admission allowance, not an estimate that every requester may independently treat as available.

Visibility, inference, and intervention authority remain separate decisions. Sharing telemetry does not authorize rescheduling a factory. Limited disclosure also limits assurance: undisclosed capacity remains unknown. A resource's digital representation must refer to the same authoritative capacity account to prevent double counting across enterprises or layers.

Federated GridOperations

GridOperations should show each channel's query rate, cost, queue age, and accepted commitments alongside combined reservations, resource demand, and settlement backlog. Enterprise teams can inspect their private state and appropriately aggregated neighborhood constraints. Access should be governed by organizational roles and scoped permissions. Physical co-location may help coordination, but is not a requirement of the architecture.

Coordination should preserve enterprise autonomy and competition. Share the minimum information needed for authorized stability functions. Operational constraints should not become a channel for coordinating prices, dividing customers, or exposing unnecessary commercial detail. Participation and intervention rules need accountable governance appropriate to each jurisdiction.

What the architecture must demonstrate

The useful test is whether separate channels plus coordinated capacity improve outcomes beyond a shared stack with existing rate limiting, queues, workflow orchestration, and resource planning. Those mechanisms already address important parts of the problem. GUDIYA's proposed contribution is continuity and coordination across the boundaries where their local views become incomplete.

A bounded pilot should compare a shared stack, isolated channel stacks with existing reservation controls, and isolated channels with Grid-assisted coordination. Apply the same human and agent workloads, including cross-channel retries, stale inventory reads, failed allowance transfers, and capacity losses. Measure customer latency, agent throughput, cost, unfair starvation, duplicate commitments, overdue obligations, and recovery time.

Also test the stabilizer. Stale forecasts, unavailable coordinators, synchronized rerouting, and conflicting interventions can create new instability. Local safeguards, expiring authority, audit records, and defined degraded modes are necessary. The Grid must demonstrate that its corrective actions reduce aggregate harm without exporting excessive burdens to participants outside its view.

The architectural conclusion

AI-Vertigo began as a concern about business functions accelerating out of coordination. Version 3 places that concern within an economy with two commerce channels: conventional digital commerce and agentic commerce can use distinct service infrastructures, while resource fulfillment and obligation accounting remain coordinated. Each layer has its own constraints and a range of operating tempos.

Stable operation depends on compatible interactions across those boundaries. The proposed Grid therefore follows Gudiya-Effects through settlement, assesses aggregate commitments against capacity, and applies the five stabilization disciplines where it has visibility and authority. The design principle is separate service envelopes with coordinated promises. Two commerce channels can support different workloads without independently claiming the same resources. Useful acceleration is the acceleration the ecosystem can absorb and fulfill, with room to recover.

Sources and editorial notes

The March 2025 AI-Vertigo origin is attributed to Ashish Warudkar’s account. The original video of this intuition is available.

[1] NOAA National Weather Service, “Turbulence.” Wind shear, atmospheric stability, and turbulence. https://www.weather.gov/source/zhu/ZHU_Training_Page/turbulence_stuff/turbulence/turbulence.htm

[2] NOAA National Weather Service, glossary entry “Kelvin-Helmholtz Waves.” <https://forecast.weather.gov/glossary.php?word=Kelvin-Helmholtz+waves>

[3] Todd Bishop, GeekWire, September 20, 2026, “Amazon blocks Meta’s Muse AI assistant in new standoff over agentic shopping.” <https://www.geekwire.com/2026/amazon-blocks-metas-muse-ai-assistant-in-new-standoff-over-agentic-shopping/>

[4] Microsoft Azure Architecture Center, Bulkhead pattern. Resource-pool isolation as a reliability pattern. <https://learn.microsoft.com/en-us/azure/architecture/patterns/bulkhead>

[5] Malcolm Featonby, Amazon Builders’ Library, Making retries safe with idempotent APIs. <https://aws.amazon.com/builders-library/making-retries-safe-with-idempotent-APIs/>

The Nadella discussion summarizes the supplied screenshot rather than independently authenticating the interview wording. GUDIYA terms, the three-layer synthesis, and cadence shear are proposed concepts. The numerical order example is hypothetical.

Figures 1–4 are illustration briefs. They require two commerce channels within the digital-services layer and a shared resource network, with coordinated reservations between them.