Why Agentic Commerce Requires State-Based Liquidity Resolution

Richard Lane & Vladan Jovanovic

Co-founders, derien

Paris, France | May 2026

This paper is the third in a series. The first paper, Machine-Native Payment Economies and the Structural Limits of Pair-Based FX Resolution (Lane, 2026), established that sequential pair-based FX architectures degrade structurally under machine-native transaction density. The second paper, Persistent Global Liquidity State (Lane & Jovanovic, 2026), examined the coordination architecture that follows. This paper addresses why autonomous systems mechanically force that architectural transition.

Abstract

Modern monetary infrastructure was designed around sequential human coordination. Currency relationships resolve independently, settlement occurs externally, and consistency between monetary states is restored indirectly through arbitrage, reconciliation, and delay. This architecture persists because human economic systems tolerate temporary inconsistency.

Autonomous systems do not.

Machine-native systems transact continuously, optimise globally, and depend on coherent monetary state at every decision point. Under these conditions, sequential monetary coordination introduces compounding state inconsistency into autonomous decision systems operating at machine speed. The result is not merely inefficiency — it is computational instability.

This paper argues that increasing autonomous transaction density transforms liquidity coordination from a market structure problem into a distributed systems problem. The requirement is no longer efficient bilateral execution. It is simultaneous global consistency across all monetary relationships — what this series has termed persistent global liquidity state.

The consequence is not an improvement to foreign exchange. It is a transition in the form of monetary coordination itself: from sequential event processing toward continuous state resolution, and from market infrastructure toward coordination infrastructure for autonomous economic systems. This transition is occurring now because autonomous systems compress economic decision cycles below the reconciliation tolerances sequential monetary architectures depend upon.

1. Introduction

Existing monetary systems evolved around assumptions that are fundamentally human. Settlement is intermittent. Decision-making is slow relative to settlement cycles. Economic coordination tolerates delay. Under these assumptions, currency relationships can resolve sequentially because inconsistencies between them can persist temporarily without destabilising the broader system.

Modern foreign exchange infrastructure operates through bilateral pair coordination: each relationship resolves independently, and consistency between pairs is not enforced directly but approximated and restored through arbitrage over time. This architecture succeeds because traditional economic systems tolerate asynchronous reconciliation. An agent making a treasury decision on Monday is not meaningfully affected by the fact that a GBP/JPY cross was inconsistent with its EUR/JPY and EUR/GBP components for thirty seconds on Tuesday.

Autonomous systems eliminate these tolerances simultaneously.

Machine-native economic systems operate continuously. They optimise globally across multi-dimensional state spaces. They transact at machine speed. And critically, they depend on economically coherent state at every decision point — not eventually, but at execution. Delayed consistency compounds into recursive optimisation instability. Inconsistent monetary state does not remain locally bounded; it propagates through every subsequent autonomous decision that depends on it.

> *Autonomous systems cannot make economically coherent decisions on top of inconsistent monetary state.*

This changes the architectural requirement entirely. The problem is no longer how currency pairs can be matched more efficiently. The problem becomes how all monetary relationships can remain globally consistent simultaneously under autonomous transaction density. This paper argues that the transition from one framing to the other is not discretionary. It is mechanically forced by the operating characteristics of autonomous economic systems.

Section 2 characterises sequential monetary coordination and its hidden assumptions. Section 3 examines why autonomous systems break those assumptions. Section 4 analyses the topology failure at scale. Section 5 argues that liquidity must resolve as state rather than as a sequence of events. Section 6 introduces economic throughput as the correct category for machine-native monetary coordination. Section 7 addresses the infrastructuralisation of monetary coordination. Section 8 concludes.

2. Sequential Monetary Coordination and Its Hidden Assumptions

Current FX infrastructure is fundamentally sequential. Currency relationships resolve independently across fragmented books, venues, and counterparties. Consistency between these relationships is not enforced directly by the system. Instead, it is restored indirectly through arbitrage, latency competition, inventory balancing, and external market-making.

This architecture creates what distributed systems would describe as eventual consistency. The system tolerates temporary divergence because human settlement is slow, human decision cycles are intermittent, and most economic systems are externally coordinated. Temporary inconsistency therefore remains economically manageable.

But the architecture contains a hidden assumption that receives insufficient attention in the market microstructure literature: decisions can continue while monetary state remains partially inconsistent. This assumption is valid under human-speed coordination. It fails structurally under autonomous coordination.

The second hidden assumption is equally important: that the actor making decisions is separate from the settlement environment and therefore insulated from its intermediate states. A human treasury manager submitting a conversion order does not depend on the real-time consistency of every other currency relationship in the system. Their decision cycle is slow enough that the arbitrage process has time to restore consistency before their next decision point.

Autonomous systems violate both assumptions simultaneously. They are not insulated from settlement state — they depend on it continuously. And their decision cycles are fast enough that sequential arbitrage reconciliation cannot restore consistency between decision points.

3. Autonomous Systems and the Decision Consistency Requirement

Autonomous systems transform monetary coordination into a state dependency problem. Unlike human systems, autonomous systems operate continuously, optimise recursively, depend on machine-speed execution, and compound decisions directly from prior system state. Under these conditions, inconsistent monetary state propagates instability into every subsequent autonomous decision.

The mechanism is specific. An autonomous procurement system managing obligations across multiple currencies must maintain a consistent view of its monetary position at every optimisation cycle. If the settlement state it observes is inconsistent — balances that have changed economically but not yet settled operationally, or cross-pair rates that are temporarily misaligned — its optimisation output will be conditioned on a false view of the world. The error does not remain local. It propagates forward into every subsequent decision that depends on the output of this one.

Under low-frequency human operations, this divergence is manageable. Under continuous autonomous optimisation, it becomes a compounding source of uncertainty. The failure modes include:

- Routing instability — optimal paths at decision point T degrade by decision point T+ε as populations of autonomous agents pursue correlated optimisation paths simultaneously.
- Treasury fragmentation — autonomous systems maintaining positions across multiple settlement environments hold inconsistent views of aggregate exposure, producing systematic over- or under-hedging.
- Recursive optimisation failure — systems whose optimisation loops assume coherent state produce incorrect outputs when state is inconsistent; those outputs become inputs to subsequent cycles, compounding the error.
- Capital duplication — prefunding requirements across fragmented settlement systems cause autonomous treasury systems to hold redundant reserves, reducing capital efficiency systematically.
- Asynchronous settlement divergence — populations of autonomous agents operating on divergent lagged views of settlement state generate collective behaviours that are difficult to predict or correct, because optimisation itself is conditioned on inconsistent information.

The mechanism behind these failure modes can be stated precisely. In human-era commerce, reconciliation lag — the interval during which monetary state is inconsistent between execution and settlement — is orders of magnitude longer than human decision cycles. Inconsistency resolves long before the next decision point arrives. The architecture is safe because decisions never compete with the inconsistency window.

Autonomous systems compress decision cycle time toward machine speed. As autonomous transaction density increases, decision cycle time falls while reconciliation lag remains structurally bounded. At some crossing point, decision cycle time falls below reconciliation lag. From that point forward, autonomous systems make every decision inside the inconsistency window. The assumption that sequential coordination relies upon does not degrade — it breaks permanently.

This crossing point is the historical trigger for the transition from sequential monetary coordination toward state-based liquidity resolution.

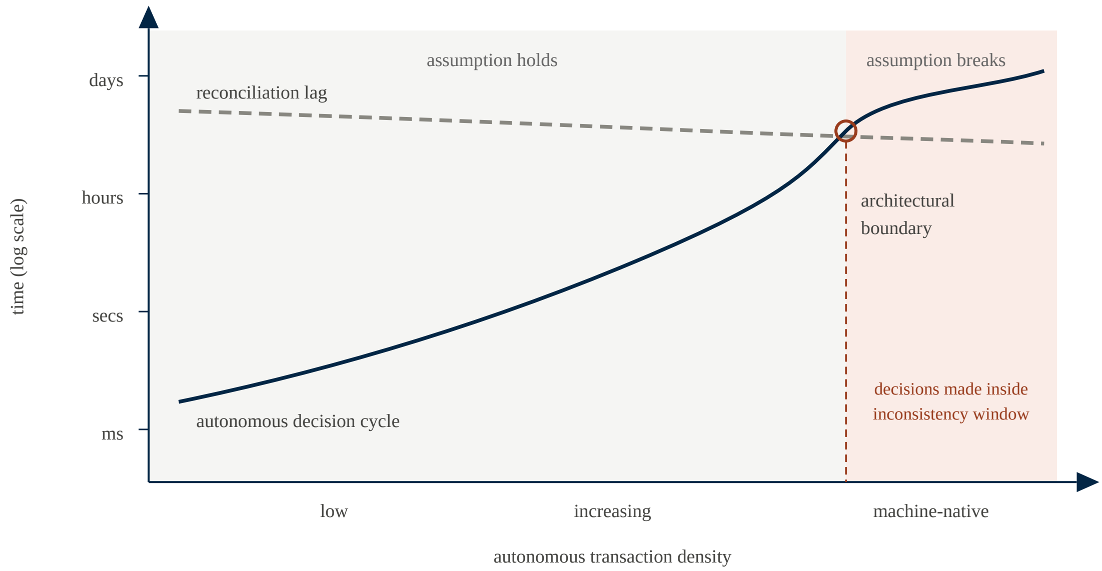

Figure 1. Reconciliation lag (dashed) remains structurally bounded as autonomous transaction density increases. Autonomous decision cycle time (solid) compresses toward machine speed. At the architectural boundary the two curves cross: autonomous systems begin making every decision inside the monetary inconsistency window. Left of the boundary the sequential coordination assumption holds. Right of it, it breaks permanently.

These are not financial inefficiencies in the conventional sense. They are computational instabilities introduced by applying sequential monetary coordination to systems whose optimisation architecture requires consistent state at every decision point. The distinction matters because it changes the category of the problem — and therefore the category of the solution.

4. The Topology Problem at Scale

Pair-based monetary architecture scales poorly under autonomous transaction density, and the scaling failure is not linear. In a system with N currencies, the number of bilateral pairs grows as $N(N-1)/2$. Triangular arbitrage paths — the minimum structure required to restore cross-pair consistency — grow as $N(N-1)(N-2)/6$.

Currencies (N)	Bilateral Pairs	Triangular Paths
10	45	120
20	190	1,140
50	1,225	19,600
100	4,950	161,700
180	16,110	955,860

Table 1. Growth of coordination complexity as a function of currency count N.

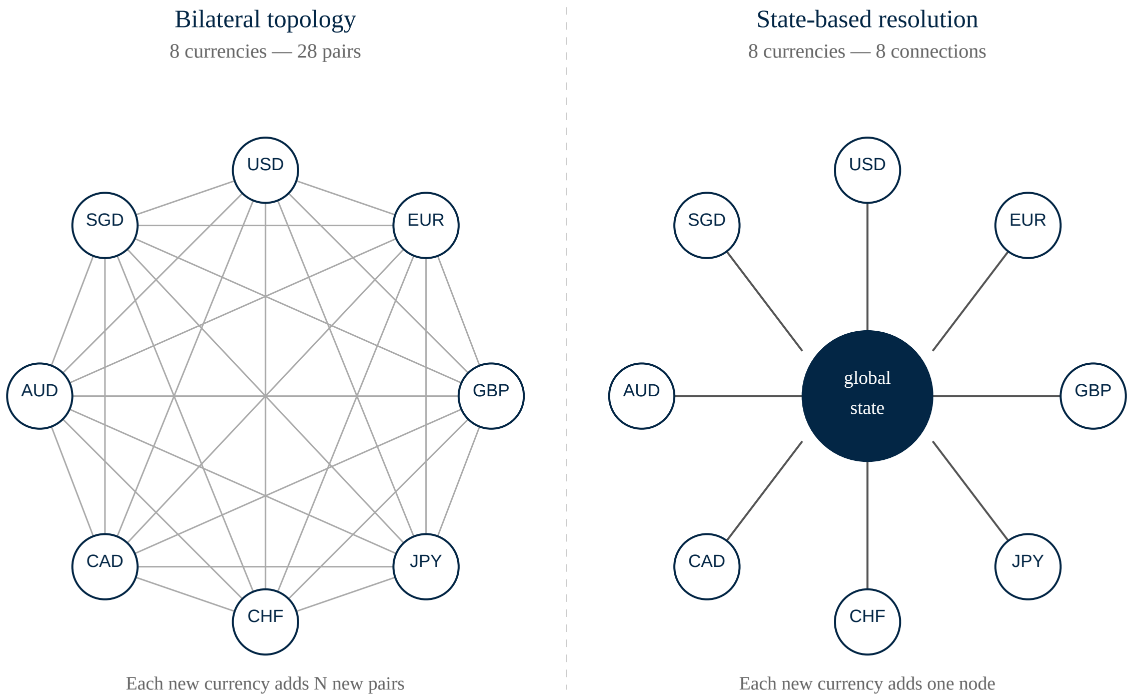

Figure 2. Left: bilateral pair topology for 8 currencies requires 28 independent connections. Right: state-based resolution requires 8 connections regardless of currency count. Adding a currency to the bilateral system adds N new pairs; adding it to the state system adds one node.

Under intermittent human transaction flow, this architecture remains viable because arbitrage has time to restore consistency between execution events. Under continuous machine-native flow, arbitrage becomes a lagging function of a continuous signal. Cross-pair inconsistencies emerge faster than reactive reconciliation can restore them.

The topology failure is therefore not simply computational inefficiency — it is a structural consequence of applying bilateral coordination topology to a genuinely multi-dimensional optimisation problem. Autonomous systems do not naturally decompose monetary relationships into bilateral pairs. They optimise across state spaces. The bilateral pair is an inherited abstraction from human-era market structure, and its limitations become progressively more visible as machine-native transaction density increases.

This is structurally similar to transitions observed in other coordination systems. Early telephone networks routed calls through a combinatorially growing number of direct bilateral connections. As network scale increased, this topology became unsustainable. The solution was not better bilateral routing. It was the invention of packet switching — a fundamentally different coordination topology that treated communication as a global routing problem rather than a collection of independent bilateral connections. Monetary coordination is approaching an equivalent boundary.

5. Liquidity as State

Under machine-native settlement environments, liquidity must resolve as state rather than as a sequence of conversion events. This claim requires precise formulation.

In sequential systems, execution, pricing, settlement, and reconciliation exist as separate economic events distributed across time. Each event produces an intermediate state that subsequent events depend on. The system is event-centric: transactions initiate resolution, liquidity is discovered reactively, and settlement occurs downstream. Consistency is restored after execution.

In simultaneous state resolution systems, these events collapse into a single consistent global monetary state. The system is state-centric: balances exist continuously within shared settlement environments, obligations evolve continuously, and feasible resolution paths emerge dynamically from aggregate system state. Consistency exists at execution rather than being restored after.

The mathematical foundation for this class of coordination problem is well established. Ford and Fulkerson (1956) demonstrated that feasible flow states could be identified within network graphs subject to capacity constraints. The contribution of machine-native payment systems is not the introduction of a new mathematical framework, but the emergence of operating conditions under which such a framework becomes structurally necessary at global scale.

Several properties emerge from simultaneous state resolution that sequential architectures cannot replicate:

- Endogenous liquidity discovery — a conversion appearing illiquid bilaterally may resolve efficiently through multi-currency closure discovered endogenously by the coordination system. Liquidity becomes a property of aggregate system state rather than isolated bilateral inventory.
- Arbitrage internalisation — because all currency relationships resolve simultaneously, cross-pair consistency is enforced at execution. Arbitrage is not required to restore consistency afterward because the system remains internally consistent by construction.
- Residual-free settlement — when resolution produces a closed state, no unresolved obligations remain outstanding after execution. The distinction between execution and settlement collapses.
- Dimensional stability — simultaneous state resolution does not scale according to the number of bilateral pairs because bilateral pairs are no longer the primary unit of computation. Adding currencies adds nodes to a global liquidity graph rather than multiplying enumerated routing paths requiring independent maintenance.

These properties are not incremental improvements to sequential coordination. They are qualitatively different architectural properties that emerge from resolving liquidity as global state rather than as a sequence of bilateral events.

6. Economic Throughput as the Correct Category

Traditional financial systems optimise execution quality, routing efficiency, spread compression, and settlement latency. These are the right optimisation targets for human-speed commerce, where the binding constraint is the cost and speed of individual transactions.

Autonomous systems optimise continuous global economic coordination. Under autonomous transaction density, the binding constraint shifts. Economic throughput — the rate at which an autonomous system can execute economically coherent decisions — becomes limited not by the speed of individual transactions, but by the system's ability to maintain coherent monetary state across distributed autonomous actors.

This is a categorical shift in what monetary coordination infrastructure is optimising for. Consider the analogy to computing infrastructure. Early computing optimised individual instruction execution speed. As systems became distributed, the binding constraint shifted from instruction speed to coordination overhead — network latency, consistency protocols, synchronisation costs. The infrastructure that mattered was no longer faster processors but coherent distributed state management.

Monetary coordination is undergoing an equivalent transition. The binding constraint for autonomous economic systems is not transaction execution speed. It is the overhead introduced by sequential monetary coordination into continuous autonomous optimisation loops. Every inconsistency requires a decision to be deferred, revised, or made on false premises. Every fragmented balance introduces redundant capital allocation. Every delayed reconciliation introduces recursive state uncertainty. These overheads are not linear — they compound across the decision graph of the autonomous system.

> *Economic throughput for autonomous systems is therefore a function of monetary state coherence, not transaction execution speed.*

This reframes monetary coordination infrastructure from market infrastructure — optimising the cost and speed of exchange events — to coordination infrastructure: optimising the coherence of monetary state across distributed autonomous decision systems.

The category shift matters because it changes who cares about this problem and why. Traditional financial infrastructure serves institutions whose primary concern is transaction cost. Coordination infrastructure serves autonomous systems whose primary concern is decision coherence. These are different customers with different requirements and different ways of evaluating solutions.

7. The Infrastructuralisation of Monetary Coordination

Every sufficiently scaled coordination layer eventually disappears into infrastructure. The pattern is consistent across coordination domains.

Networking did not improve incrementally. Point-to-point communication networks faced combinatorial scaling failures as the number of nodes grew. The solution was not better bilateral connections. It was the invention of packet switching and globally coordinated routing infrastructure — a fundamentally different coordination topology that made the underlying network invisible to the applications built on top of it.

Computation did not improve incrementally. Isolated local systems faced scaling limits that required distributed coordination. The solution was not faster local processors. It was the development of distributed computing infrastructure — consistent state management, coordination protocols, and eventually cloud utility infrastructure that made compute capacity invisible to the applications built on top of it.

Monetary coordination appears to be approaching the same boundary. Sequential pair-based FX infrastructure was designed for a world of intermittent human transactions. Autonomous economic systems operate at a qualitatively different scale and with qualitatively different requirements. The solution will not be faster bilateral matching or cheaper correspondent banking.

The systems that recognise this transition earliest will not build better exchanges. They will build on top of coordination layers that make monetary exchange invisible — infrastructure beneath autonomous economic systems in the same way that TCP/IP is infrastructure beneath internet applications.

The consequence of this transition is not improved FX infrastructure. It is the emergence of a new infrastructure category: coordination infrastructure for machine-native economic systems. This category is not defined by what it replaces — foreign exchange markets, correspondent banking, settlement networks — but by what it enables: continuous autonomous economic activity operating on coherent global monetary state.

8. Conclusion

Sequential monetary coordination was designed for economies where decisions were slower than reconciliation. The assumptions that made sequential coordination viable — delayed consistency, external settlement, and human-speed decision cycles — are no longer satisfied by autonomous economic systems.

This transforms liquidity coordination from a market structure problem into a distributed systems requirement. The failure modes introduced by sequential coordination into autonomous systems — routing instability, treasury fragmentation, recursive optimisation failure, capital duplication, asynchronous settlement divergence — are not financial inefficiencies. They are computational instabilities.

Under sufficiently dense machine-native economies, liquidity must therefore resolve as simultaneous global state. The binding constraint for autonomous economic systems is not transaction cost but decision coherence, and decision coherence requires consistent monetary state at every decision point — not eventually, but at execution.

The consequence is not a better FX market. It is a transition in the architecture of monetary coordination itself: from sequential event processing toward continuous state resolution, from market infrastructure toward coordination infrastructure, and from exchange as an economic event toward exchange as invisible infrastructure beneath autonomous economic systems.

Monetary coordination is approaching the same boundary that networking and computation crossed before it. The coordination layer that makes monetary exchange invisible will become foundational infrastructure for machine-native economic systems — not because it is superior to what it replaces, but because the operating conditions of autonomous commerce make sequential coordination architecturally incompatible with the systems they must serve.

References

Brewer, Eric (2000). Towards Robust Distributed Systems. Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC).

Ford, L.R.; Fulkerson, D.R. (1956). Maximal Flow Through a Network. Canadian Journal of Mathematics, 8, 399–404.

Lamport, Leslie (1978). Time, Clocks, and the Ordering of Events in a Distributed System. Communications of the ACM, 21(7), 558–565.

Lane, Richard (2026). Machine-Native Payment Economies and the Structural Limits of Pair-Based FX Resolution. Working Paper, derien.

Lane, Richard; Jovanovic, Vladan (2026). Persistent Global Liquidity State. Working Paper, derien.

Lynch, Nancy (1996). Distributed Algorithms. Morgan Kaufmann.

Lyons, Richard (2001). The Microstructure Approach to Exchange Rates. MIT Press.

Nakamoto, Satoshi (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. bitcoin.org.

O'Hara, Maureen (1995). Market Microstructure Theory. Blackwell.

Tanenbaum, Andrew; van Steen, Maarten (2007). Distributed Systems: Principles and Paradigms. Prentice Hall.