International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.4 (2025) pp. 8810-8820
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Agentic Commerce: The Paradigm Shift from Human-Mediated to Autonomous AI-Driven Transactions in Digital Payment Systems

Krishna Dusad*

University of Illinois, Urbana Champaign-USA
* **Corresponding Author Email:** krishnadusad27@gmail.com- **ORCID:** 0000-0002-1247-7850

Article Info:

DOI: 10.22399/ijcesen.4304
Received : 22 September 2025
Revised : 09 November 2025
Accepted : 11 November 2025

Keywords

agentic commerce,
autonomous transactions,
AI payment systems,
fintech convergence,
LLM-based agents

Abstract:

Agentic commerce represents a monumental change from human-to-human digital commerce in that the buying decisions that humans used to make are being supplanted in the future with automated and self-aware AI-driven commerce. The advent of agents based on large language models can now, to varying degrees, engage with users through relatively natural conversation to learn what they require, search across multiple platforms, negotiate prices, and autonomously handle the entire transaction process without human involvement. As a result, a completely new payment construct, authentication method, and security model are needed. Traditional banks and major technology companies, payment processors, card networks, and AI research organizations have begun working together to build the technical architecture that will allow agents to autonomously engage in commercial transactions. Numerous issues need to be resolved, including the appropriate way to verify AI agents acting on behalf of users, formalizing dispute resolution procedures, enforcing consumer protections, and addressing ethical concerns with machines making judgments about purchases on behalf of humans. The union of AI and fintech is an exciting space that will likely increase convenience and productivity, while demanding governments adapt their regulatory frameworks and the commercial ecosystem create trust that also fits the compliance model of the regulatory body in place.

1. The Rise of Machine-Driven Transaction Systems in Modern Financial Technology

1.1 Understanding Machine-Operated Commercial Systems

Machine-operated commercial systems have emerged where software programs complete buying activities without human involvement. These programs analyze what people need, locate suitable products, handle price discussions, and finish payment processes independently [1]. Modern language processing technology enables these programs to interpret natural conversations and translate them into concrete purchasing actions. Rather than serving merely as search tools or recommendation engines, these systems possess decision-making capabilities that allow them to bind legal agreements and transfer funds. Financial institutions now recognize these software entities as legitimate transaction initiators within established payment networks.

1.2 Evolution from Desktop Shopping to Machine-Initiated Purchases

Commercial technology has undergone successive transformations since online shopping first appeared. Initial web stores required customers to browse catalogs and manually enter payment details on desktop computers. Smartphone technology later introduced portable purchasing, enabling buyers to complete transactions anywhere while adding features like geographic targeting and tap-to-pay functions [2]. Every technology advancement has had a profound impact on merchant operations and consumer purchasing habits. Current developments introduce thinking machines that initiate and complete purchases independently, representing a qualitative change from previous innovations where humans retained control over each transaction step.

1.3 Key Issues and Transaction Automation's Effects

Transaction automation through intelligent software raises fundamental questions about market participation and economic decision-making. This development affects technical standards, business operations, and regulatory frameworks simultaneously [1]. Financial networks must create new protocols for verifying machine identities, while legal systems need frameworks for disputes involving software-initiated purchases. Market dynamics shift when machines negotiate prices and select products based on algorithmic logic rather than human preferences. These changes require examination across multiple domains to understand their full ramifications.

1.4 Major Participants Building Machine Commerce Infrastructure

Multiple organizations contribute different components needed for machine-initiated transactions to function reliably. Card processing companies, including Visa and Mastercard, have developed standards that allow machines to access payment rails previously designed for human cardholders. Technology firms like Stripe and PayPal create programming interfaces specifically for machine authentication and transaction processing. Artificial intelligence developers at organizations such as Google and Anthropic build the underlying decision-making capabilities these transaction systems require. Initial deployments focus on regular purchases like subscriptions and reorders, gradually expanding capabilities toward complex negotiations [1].

2. Conceptual Foundations: The Shift from Manual to Automated Commercial Processes

2.1 Conventional Digital Marketplace Structures and Human Navigation Patterns

Digital marketplaces emerged with particular beliefs about how people shop online. These platforms arrange themselves around visual form, categorical hierarchy, and sequential checkout, similar to the layout of physical stores [3]. Consumers will move through each stage in a predictable manner: Discovery, Evaluation, Selection, and Completion of Payment. Technical components such as recommendation engines, wish lists, and reminders for abandoned cart items all presume that a person is behind the screen, and navigating the exhibition using visual-based choices with influence from images, reviews, and prices to compare. The security protocols make this assumption as well; they analyze human behavior, including typing habits, mouse trajectory, window size, duration of session, etc., to separate valid consumers from potential automated attacks. This dedication to human-in-the-loop design philosophy has informed two decades of online retail development.

2.2 Language Processing Systems Operating as Independent Market Participants

Advanced text comprehension programs now operate as independent buyers that are capable of purchasing within commercial networks. They take spoken or written instructions and create executable purchase orders without any human intervention. The real transformation arises when software interpreters command phrases like, "... find me the cheapest flights next month..." and independently purchase tickets after checking availability with respective airlines. These programs can retain persistent goals, set spending limits, bring and apply decision criteria consistently across multiple purchase transactions, and include some form of logging. To effectively process transactions, payment platforms now must distinguish between a person pushing the "buy" button and software pushing the "buy" button through programming interfaces. It does matter in the context of fraud prevention, dispute resolution, and legal liability in the event a transaction goes poorly.

2.3 Contrasting Decision Patterns Between People and Software Purchasers

People shop with inconsistency—sometimes methodical, often spontaneous. Brand memories from childhood influence choices, mood affects spending, and social pressures drive purchases. Software operates differently, following predetermined rules without deviation [4]. Where someone might buy expensive coffee despite cheaper alternatives because they enjoy the café atmosphere, programs are selected based solely on programmed parameters. This behavioral gulf creates marketplace disruptions. Dynamic pricing algorithms designed to exploit human urgency fail against patient software. Recommendation systems trained on human browsing patterns produce irrelevant suggestions for goal-oriented programs. Retailers must reconsider fundamental assumptions about buyer motivation and decision timing.

2.4 Market Structure Changes When Software Handles Transactions Directly

Direct software purchases shift competition in many industries. Because programs search all

suppliers at once, the "price discovery" process occurs instantly. There are no more geographic advantages, and there are no information gaps that allowed traditional retail margins [3]. Small vendors can access customers through aggregation platforms, and well-known brand perception advantages that human-to-human behavior encourages can displace the human buyers. The legacy employment types of sales, purchasing departments, and customer service departments are at serious risk of obsolescence. New employable specializations emerge around teaching software, how to shop effectively, and monitoring performance. The regulatory systems designed for human-to-human commerce need to be modified with respect to who is responsible for unauthorized purchases from autonomous software. Who decides on returns when no human decides which item to order? All of these structural changes will affect supply chains, payment systems, and legal systems.

2.5 Technical Architecture Deep Dive: The Engineering Reality Behind Agent Communication

The Model Context Protocol (MCP) emerges from fundamental limitations in how software systems have historically communicated. Traditional REST APIs operate on a principle of statelessness: each request stands alone, carrying no memory of previous interactions. This design philosophy, while elegant for simple web applications, creates inefficiencies when applied to intelligent agents that benefit from persistent context and accumulated understanding.MCP addresses these limitations through a fundamentally different architectural approach. Rather than treating each interaction as an isolated event, MCP maintains session-based context that allows agents to build upon previous exchanges [11]. When an agent queries a payment system through MCP, it retains memory of earlier transactions, user preferences, and contextual details that inform subsequent decisions. This persistent state management represents a significant departure from the rebuild-and-resend approach required by traditional APIs.The protocol's technical implementation reveals sophisticated optimizations specifically designed for AI workloads. Context compression techniques can reduce data transmission by up to seventy percent compared to stateless alternatives, while real-time streaming capabilities enable sub-100-millisecond response times for context operations. These performance characteristics become critical when agents orchestrate complex, multi-step commercial processes where latency compounds across multiple system interactions [11].Perhaps most significantly, MCP introduces dynamic capability discovery that fundamentally changes how software systems integrate. Rather than requiring developers to study documentation and hard-code API endpoints, agents can query MCP servers directly about available capabilities. An agent connecting to Stripe's MCP server, for instance, can ask what payment operations are available and receive structured responses detailing everything from customer creation to refund processing. This dynamic discovery enables agents to adapt to new services and capabilities without requiring updates to their core logic.

2.5.1 Agent-to-Agent Protocol: Decentralized Coordination Architecture

Google's approach with the Agent-to-Agent protocol tackles a different but equally critical challenge: enabling independent AI agents to discover, communicate with, and coordinate tasks among themselves [12]. While MCP focuses on the agent-to-system relationship, A2A addresses the emerging need for agent-to-agent collaboration in complex commercial scenarios.The A2A architecture builds upon established web standards while introducing agent-specific enhancements. At its foundation lies the concept of Agent Cards standardized JSON files hosted at predictable web locations that function as digital business cards for AI agents [12]. These cards contain essential metadata including the agent's capabilities, security requirements, supported data formats, and communication endpoints. The standardization of this discovery mechanism allows any A2A-compatible agent to identify and evaluate potential collaborators without prior knowledge of their existence.Communication within A2A follows familiar web protocols JSON-RPC over HTTPS but extends these foundations with agent-specific message structures and task lifecycle management. Tasks become first-class entities with unique identifiers and defined state transitions, enabling agents to coordinate complex workflows that might span hours or days [12]. The protocol's support for asynchronous communication through webhooks and Server-Sent Events accommodates the reality that agent-driven processes often unfold over extended timeframes.Security in A2A reflects the protocol's distributed nature through mutual authentication requirements. Unlike traditional client-server models where the server validates the client, A2A agents must verify each other's identities and credentials before engaging in collaboration. This peer-to-peer security model supports OAuth 2.0 with PKCE extensions and API key authentication, providing flexible options for different deployment scenarios.

2.5.2 Architectural Tensions and Design Philosophy Differences

The fundamental architectural differences between MCP and A2A reflect distinct design philosophies about how agentic systems should operate. MCP embodies a hub-and-spoke model where agents connect to centralized resources and services [11]. This approach optimizes for performance and simplicity in agent development. Developers can focus on agent logic while relying on MCP servers to handle the complexities of system integration.A2A, conversely, embraces a decentralized mesh architecture where agents operate as peers in a distributed network [12]. This design prioritizes autonomy and resilience over centralized optimization. Agents in an A2A network can continue operating even if individual nodes fail, and new agents can join the network without requiring approval from central authorities.These architectural differences create interesting tensions in real-world implementations. MCP's centralized approach enables sophisticated optimizations and coordinated resource management, but potentially creates bottlenecks and single points of failure. A2A's decentralized model offers greater resilience and autonomy but requires agents to handle more complexity in discovering and coordinating with peers.The performance characteristics of each protocol reflect these design choices. MCP excels at high-frequency, low-latency interactions between agents and tools, processing thousands of operations per second with consistent response times. A2A prioritizes reliable message delivery and coordination across potentially unreliable network conditions, optimizing for eventual consistency rather than immediate response.

2.5.3 Integration Patterns in Practice: Hybrid Architectures for Complex Commerce

The most sophisticated agentic commerce systems emerging today combine both protocols, leveraging each for its strengths while mitigating individual limitations. In these hybrid architectures, MCP handles the direct interface between agents and business systems, payment processors, inventory databases, customer relationship management platforms while A2A manages coordination between specialized agents with complementary capabilities [11][12].Consider a complex procurement scenario where an enterprise agent must coordinate with supplier agents to negotiate bulk purchasing agreements. The enterprise agent uses MCP to access internal systems budget databases, approval workflows, inventory requirements gathering the context necessary for informed decision-making. Simultaneously, it employs A2A protocols to discover and negotiate with supplier agents, each representing different vendors with distinct capabilities and pricing models.This layered approach addresses scalability challenges that neither protocol could handle alone. MCP provides the high-performance, context-rich access to business systems that agents require for intelligent decision-making, while A2A enables the distributed coordination necessary for complex multi-party negotiations that characterize sophisticated commercial relationships.The technical implementation of such hybrid systems reveals emerging patterns in agent architecture. Successful implementations typically separate concerns between system integration capabilities handled through MCP connections and inter-agent communication capabilities managed through A2A interfaces. This separation allows agents to specialize in their core competencies while relying on standardized protocols for integration and coordination.

2.5.4 Google's Agent Payments Protocol (AP2): mandates, push/pull coverage, and roles

Google's **Agent Payments Protocol (AP2)** is an open, payment-method-agnostic extension designed to make **agent-initiated purchases** verifiable and interoperable. AP2 builds on **A2A** for agent-to-agent messaging and is designed to co-exist with **MCP** for agent-to-tool integrations. Its core design introduces **verifiable digital credentials (VDCs)** called **mandates** that turn a purchase into a signed, auditable contract rather than a one-off API call

AP2 mandate types (VDCs)

- **Cart Mandate (human-present):** Created by the *merchant*, then **cryptographically signed by the user** on a trusted surface. Binds identity, exact cart contents, amount/currency, delivery details, and a risk payload. Evidence for representment. AP2 Protocol
- **Intent Mandate (human-not-present):** **User-signed** authorization capturing *the agent's restated understanding of the user's instruction*, budget/limits, TTL, allowed payees/categories, and risk payload. Enables autonomous purchases within bounds when the user is away. AP2 Protocol
- **Payment Mandate (network/issuer visibility):** A separate credential bound to the Cart/Intent mandates that signals **agent**

involvement and modality (human-present vs not-present) to the payment network/issuer; portions may be shared (with consent) for risk control and later **used as dispute evidence**

Human-present vs human-not-present flows AP2 standardizes both modalities. In human-not-present mode, the merchant may **force step-up** to bring the user back in-session if confidence is low (e.g., SKU selection or Q&A), upgrading an intent into a cart mandate. AP2 Protocol

Push vs. pull coverage

- **Initial focus: pull** methods (e.g., card rails), including merchant/user-initiated step-up challenges and network visibility. AP2 Protocol
- **Roadmap: push** transfers (e.g., UPI, Pix), wallets, and digital assets, keeping AP2 payment-method agnostic. AP2 Protocol For broader context, *push* = payer sends money (one-offs, instant banking); *pull* = payee debits with an authorized **mandate** (subscriptions/VRP).

AP2 × A2A × MCP fit

AP2 defines the **payment contracts and audit trail**; **A2A** carries AP2 messages/artifacts between agents; **MCP** remains the way agents call tools and enterprise systems. Google publishes an **A2A extension** describing how Cart/Intent/Payment mandates are embedded in A2A messages/artifacts and how agents advertise their AP2 roles (merchant, shopper, credentials-provider, payment-processor).

Why AP2 matters

AP2 anchors agent commerce to **deterministic, non-repudiable proof of user intent** and a **cryptographic audit trail** for all parties—user, merchant, network/issuer—closing key trust and liability gaps left by agent "hallucinations" or misinterpretations.

3. Building Blocks for Machine-Executed Payments

3.1 Why Today's Payment Rails Struggle with Software Buyers

Modern payment systems emerged when only humans made purchases. Every security feature assumes a person sits at the keyboard, from entering memorable passwords to receiving text message codes on phones. Banks flag unusual activity by comparing it against typical human spending, catching fraudsters, and blocking software that buys faster than people ever could [5]. Merchants limit how many items ship to one address, suspecting bulk resellers rather than recognizing legitimate programs consolidating orders. Transaction systems throttle rapid requests, interpreting machine efficiency as attack behavior. Even simple tasks like entering billing addresses become obstacles when software lacks fingers to type or eyes to read security images. These friction points reveal how deeply human assumptions embed themselves in financial infrastructure.

3.2 Creating Identity Systems for Software That Spends Money

Software needs identification just as people carry driver's licenses, but digital identity works differently. Cryptographic credentials, which are mathematical proofs of identification that replace passwords that no software could remember anyway, are provided to programs by new frameworks [6]. Rather than all-or-nothing access, these systems partition permissions carefully. A grocery agent might access funds only at supermarket merchants during preset hours, while travel software activates solely for airline and hotel bookings. Some protocols require dual control, where software proposes purchases but waits for human confirmation on expensive items. Emergency shutoffs let users instantly revoke agent permissions, preventing damage from compromised systems. Building trust means creating audit trails that show exactly which agent spent what amount where, providing accountability without requiring human-like authentication.

3.3 Making Different Payment Systems Speak the Same Language

Software struggles when every payment company invents unique ways to process transactions. One provider might label shipping information as "delivery_address" while another calls it "recipient_location," forcing agents to translate constantly between systems [5]. Currency codes differ, date formats vary, and error messages come in formats that aren't compatible. Standardization attempts focus on creating common vocabularies that all providers understand. This means creating universal codes for reasons for decline and figuring out if "payment_complete" and "transaction_successful" are interchangeable terms. Beyond technical specifications, semantic alignment ensures agents interpret business concepts consistently, distinguishing between

refunds and chargebacks, understanding partial shipments, and recognizing when taxes apply.

3.4 How Major Platforms Enable Software Shopping Today

Payment companies tackle agent commerce through distinct strategies. Leading payment processors built separate endpoints where registered programs obtain special access tokens, replacing browser cookies that software cannot manage. They restructured rate limits around agent behavior, allowing rapid catalog queries while maintaining fraud protections on actual purchases [6]. Major digital payment platforms created testing grounds where developers verify agent logic before touching real money, adding dispute flows that programs navigate without human intervention. Card networks issue single-use numbers for each agent transaction, tracking exactly which software charged what amount. Some generate virtual cards that are valid only at specific merchants or during narrow time windows. These varied approaches highlight ongoing experimentation as companies balance innovation against risk, learning which freedoms agents need while maintaining security standards that protect everyone's money.

3.5 AP2 mandates and open-banking “push vs. pull” (context and interoperability)

Push A2A (payer-initiated bank transfers) are ideal for instant one-offs; **Pull A2A** relies on an **ongoing mandate/consent** for merchant-initiated debits (e.g., subscriptions, UK **VRP**). AP2’s roadmap explicitly contemplates both, so the same cryptographic **mandate primitives** can front cards today and power **UPI/Pix/VRP** in future releases. This allows consistent **authorization semantics** (who said what, when, and within which limits) regardless of the underlying rail.

4. When Money Tech Meets Machine Intelligence: A New Industry Forms

4.1 Why Banks Need Tech Labs and Tech Labs Need Banks

Technology companies excel at building smart software but stumble when navigating banking rules. Financial institutions understand money movement but cannot create cutting-edge language models. Each side holds pieces of a puzzle that neither can complete alone [7]. Banks bring decades of regulatory knowledge, risk management systems, and millions of customers who already trust them with money. Tech laboratories offer computational power, algorithm design, and teams that push boundaries of what machines can do. Partnerships form because isolation means irrelevance. A bank trying to build language models wastes resources competing against specialized labs. Tech companies attempting payment processing face regulatory mazes that banks navigate daily. Collaborative initiatives range from sophisticated platforms where machines manage whole banking relationships to more limited fraud-detection technologies.

4.2 Card networks and IT giants stake their claims.

Each major company picks a different battlefield in this emerging landscape. Leading cloud service providers turn their infrastructure into testing grounds for financial agents, betting that scale matters most. Specialized AI laboratories take another path, building models that prioritize safety over speed because financial mistakes cost real money [8]. Major card networks respond by creating special identification systems just for software buyers, like social security numbers for machines that shop. Global payment network operators rebuild parts of their infrastructure, knowing that card numbers designed for plastic rectangles make little sense for programs. These companies bring different strengths. Cloud providers own infrastructure that processes billions of requests. AI safety-focused labs attract teams focused on making reliable systems. Card networks connect to merchants everywhere. Global payment processors operate networks spanning continents. Success requires picking the right focus area and executing better than competitors with similar ideas.

4.3 Where Money Flows: Tells the Real Story

Venture funding maps which ideas investors consider worthwhile. Initial capital went to plumbing companies building connectors between payment systems and agent platforms [7]. Without these bridges, nothing else functions. Money now chases companies creating specialized shopping agents. One startup builds agents that book complicated travel itineraries. Another focuses on agents that manage household subscriptions. Traditional measurements are confused by valuations since revenue is dependent on adoption curves that are impossible to anticipate. Big companies buy smaller ones possessing key technologies or important partnerships. Silicon

Valley leads initially, but other regions catch up as governments clarify rules. Singapore attracts companies serving Asian markets. London becomes a hub for firms targeting European regulations. Investment patterns suggest believers outnumber skeptics, though skeptics point to previous technology bubbles as warnings.

4.4 Governments Scramble to Write Rules for Machine Commerce

Legal accounts are predicated on the assumption that individuals will be making decisions on their behalf; there are gaps in the law when a computer system-agent acts on their behalf. Who is liable when an agent purchases something that the person did not want? The law currently provides no clear answers to these questions [8]. Some countries are rushing to develop new classes of transactions for transactions involving agents, while others are desperately trying to squeeze new technologies into existing regulatory boxes, defining agents in one moment, payment instruments the next, and unregulated software by the political winds. Anti-money laundering rules need to be reimagined when machines engage in jointly issued digital currency transfers in patterns that no human would have considered. Consumer protection (cooling off periods, returns policies designed for human psychology) has yet to develop for transactions involving agents. Countries are beginning to realize that overly restrictive limitations move innovation elsewhere, and creating no limitations may lead to financial disaster. Sandbox opportunities give companies the freedom to discover within boundaries, while regulators get the opportunity to understand what works before creating permanent rules.

5. Obstacles and Safety Measures in Machine-Controlled Purchasing

5.1 Weak Points Where Automated Buyers Get Compromised

Criminals find new ways to exploit programs that handle money, creating risks unlike anything seen with credit card fraud. These attackers poison the instructions agents follow, making them buy worthless products at extreme prices or send payments to fake merchants [9]. Some hackers slip malicious code into agent memory, stealing payment details that get reused thousands of times before anyone notices. Network intercepts catch transactions mid-journey, changing destination accounts while keeping amounts identical to avoid detection. The speed problem makes everything worse—a corrupted agent might complete hundreds of bad transactions before morning coffee. Connected agent systems spread problems like viruses, where breaking into one means accessing many. Protection needs multiple walls: scrambling data, watching for strange behavior, and hard limits preventing catastrophic losses even when other defenses fail.

5.2 Proving Which Machine Has Permission to Spend

Making sure the right software accesses the right money sounds simple until implementation begins. People remember passwords, recognize faces, and have fingerprints—machines have none of these [10]. Instead, mathematical signatures serve as identity cards that cannot be forged or forgotten. Time locks ensure certain agents only work during business hours, while location checks confirm they run from approved servers. Merchant restrictions mean grocery agents cannot suddenly book flights, due to damage from hijacked systems. Regular check-ins force agents to prove they still operate under legitimate control, shutting down orphaned processes. Building trust means creating trails showing exactly which version of which agent did what, providing evidence when something goes sideways.

5.3 Untangling Messes When Machines Buy Wrong Things

Arguments about bad purchases get complicated when neither buyer nor seller is human. Traditional refund processes expect someone to explain what went wrong, but agents cannot testify about their reasoning [9]. Determining blame requires new thinking: did the user give unclear instructions, did the agent misinterpret reasonable commands, or did merchants mislead automated buyers? Time limits designed around human attention spans expire while users remain unaware that their agents bought anything. Evidence takes new forms—instruction logs, decision trees, and processing records replace human memory. Insurance companies scramble to price coverage for risks they barely understand, knowing that one bad algorithm could trigger thousands of claims simultaneously.

5.4 Making People Comfortable Letting Machines Shop

Fear of losing money to runaway software keeps many from trying agent shopping, requiring careful trust-building approaches. Clear displays of agent activities help without drowning users in details

they cannot process [10]. Simple controls let people set boundaries using everyday language instead of programming terms. Alert systems need balance—too many notifications get ignored, too few leave users feeling abandoned. Bad experiences spread quickly through social networks, making early mistakes especially costly for adoption. Teaching materials must explain capabilities honestly without overwhelming newcomers. Quality marks could help users identify safer platforms, though defining standards remains contentious. Backup plans matter most—users need confidence they can recover when agents misbehave.

5.5 What Takes Place When Values Are Judged by Machines

Encoding shopping preferences forces uncomfortable questions about whose values matter. Price-focused agents might fund sweatshops, while speed-optimized systems choose air freight over cleaner alternatives [9]. Teaching machines about ethics proves harder than teaching them about products. Coordinated agents could corner markets on scarce goods, creating artificial shortages that harm human buyers. Wealthy users might afford sophisticated agents that consistently beat basic versions, widening economic gaps. Society must decide whether shopping algorithms should consider only individual benefit or broader community impact.

5.6 Disputes, risk signals, and step-up in AP2

Dispute evidence model.

AP2 treats Cart/Intent mandates as signed, immutable JSON artifacts that memorialize exactly what was authorized by whom, when, and under what terms. During chargeback/representment, merchants can furnish these artifacts (plus attestation/public key) to adjudicators and networks/issuers. Payment Mandate gives the network/issuer verifiable agent-presence and modality cues, improving fraud decisions and post-event analysis. AP2 Protocol

Risk-signal envelope. AP2 includes an extensible **risk payload** (device, timing, agent identity, mandate-merchant matching) recognizing novel agentic risks such as **asynchronicity**, **delegated trust**, and **temporal gaps** between tokenization and execution. These signals are intentionally open-ended so issuers/processors can evolve models without fragmenting the protocol. AP2 Protocol

Challenges / step-up. Any party (issuer, merchant, credential provider) can **challenge** via existing rails (e.g., 3DS2). In human-not-present scenarios, merchants can **force user re-entry** before fulfillment—either to confirm SKUs (generating a Cart Mandate) or to enrich the Intent Mandate—balancing conversion vs. liability. AP2 Protocol

5.7 Proposed mandate extensions (for standardization and research)

AP2 defines three mandates today. For agentic commerce to cover recurring, merchant-initiated, and post-purchase flows across push/pull rails, your paper can propose these **forward-compatible** mandate types (all as VDCs consistent with AP2's model):

1. **Merchant Mandate (seller obligations & adjustments):**
   A **merchant-signed** credential that declares *merchant commitments* (SLA windows, refundability class, delivery constraints), and supports **merchant-initiated adjustments** (partial refunds, backorder substitutions) with user-visible cryptographic linkage to the original Cart/Intent. This formalizes the "merchant signature" AP2 already requires for Cart Mandates into a first-class, re-presentable artifact. AP2 Protocol
2. **Series (Recurring / VRP) Mandate:**
   A **user-signed** umbrella mandate for **variable recurring payments** with caps (per-charge, frequency, lifetime), merchant allow-lists, and cancellation semantics. Maps to **open-banking VRP** in pull A2A, and to card MIT/credential-on-file rules on pull rails. The Paypers
3. **Push-Authorization Mandate:**
   A **user-signed** mandate that pre-authorizes **agent-initiated push transfers** (UPI Autopay/Pix Cobrança–style) with dynamic linking to payee and constraints (amount/FX/expiry). Aligns with AP2's push roadmap while preserving the same dispute/audit model. AP2 Protocol
4. **Post-Purchase Resolution Mandate:**
   A bilateral mandate (user + merchant) that codifies **automated dispute workflows**: acceptable remedies (refund, replacement, store credit), evidence required, and time-bounds—allowing **agents** to execute resolution without human escalation in common cases, while preserving appeal paths.

AP2 in one diagram

1. **User ↔ Agent**: user signs an **Intent** (HNP) or **Cart** (HP) mandate.
2. **Agent ↔ Merchant**: merchant signs and returns Cart; agent prepares **Payment Mandate**.
3. **Merchant/PSP ↔ Network/Issuer**: Payment Mandate appended to auth → risk & agent-presence signals flow; disputes later reference these artifacts.

Table 1: *Evolution of Digital Commerce Paradigms [1, 2]*

Commerce Era	Key Characteristics	User Interaction Model	Technology Foundation	Time Period
Desktop E-commerce	Fixed location purchases, Manual browsing, HTML-based interfaces	Click-through navigation, Form-based input	Web browsers, Payment gateways	1995-2007
Mobile Commerce	Location-independent, Context-aware services, App-based transactions	Touch interfaces, Biometric authentication	Smartphones, NFC/QR codes	2008-2020
Agent-Autonomous Commerce	Machine-initiated transactions, Natural language instructions, Multi-platform execution	Conversational commands, Delegated authority	LLMs, API ecosystems	2021-Present

Table 2: *Comparison of Human-Driven vs Agent-Driven Transaction Characteristics [3, 4]*

Transaction Aspect	Human-Driven Commerce	Agent-Driven Commerce
Decision Factors	Emotions, brand loyalty, aesthetics, and social influence	Programmed parameters, logical optimization, specified criteria
Processing Speed	Variable, often slow, subject to distraction	Consistent, rapid, systematic evaluation
Information Processing	Limited comparison capacity, visual-based	Comprehensive database scanning, data-based
Purchase Patterns	Impulsive, inconsistent, experience-influenced	Predictable, rule-based, goal-oriented
Error Types	Forgetfulness, emotional decisions, and misunderstandings	Algorithmic misinterpretation, specification errors

Table 3: *AP2 Protocol Mandate Specifications [11, 12]*

Mandate	Who signs	When used	Scope & key fields	Typical rails	Evidence in disputes
Cart Mandate	Merchant signs cart; **User signs approval**	**Human-present** checkout	Exact SKUs, amount, address, risk payload	Pull today; push later	Yes (user- and merchant-signed JSON) (AP2 Protocol)
Intent Mandate	**User signs**	**Human-not-present**; delegated autonomy	Natural-language instruction playback, budget, TTL, allowed methods/merchants, risk	Pull today; push later	Yes (user-signed JSON) (AP2 Protocol)
Payment Mandate	Agent/wallet constructs; shared to network/issuer	Both modalities	Agent presence flag; modality; hash linkage; optional fields (with consent)	Appended to auth messages	Yes (network/issuer context) (AP2 Protocol)

Table 4: *Major Stakeholder Contributions to Agent Commerce Infrastructure [7, 8]*

Stakeholder Category	Key Players	Primary Contribution	Strategic Focus
AI Technology Labs	Leading cloud providers, Specialized AI labs	Language models, Cloud infrastructure, Safety protocols	Model development, Scalable computing
Payment Networks	Major card networks, Global payment processors	Transaction protocols, Tokenization systems, Network adaptations	Agent identification, Secure routing
Fintech Platforms	Payment processors, Digital payment platforms	API development, Authentication systems, Testing environments	Developer tools, Integration services
Traditional Banks	Various financial institutions	Regulatory expertise, Customer relationships, Risk frameworks	Compliance, Trust building

Table 5: *Security Vulnerabilities and Mitigation Strategies in Agent Systems [9, 10]*

Vulnerability Type	Attack Vector	Potential Impact	Mitigation Strategy
Credential Compromise	Memory exploits, Token theft	Unauthorized spending, Account drainage	Hardware-backed attestation, Encrypted storage
Instruction Manipulation	Injection attacks, Command poisoning	Misdirected purchases, Price manipulation	Input validation, Behavioral monitoring
Network Interception	Man-in-the-middle, Redirect attacks	Transaction hijacking, Destination changes	End-to-end encryption, Certificate pinning
System Cascades	Connected agent breaches, Viral spread	Mass compromise, Systemic failure	Network segmentation, Circuit breakers
Authentication Bypass	Identity spoofing, Authorization forgery	Impersonation, Illegal access	Multi-factor verification, Time-based restrictions

6. Conclusions

Agent-autonomous commerce represents a core transformation in digital commerce, the ability for software apps, or agents, to communicate on behalf of their human creators or operators, freely move throughout digital markets, negotiate terms, and transact without human involvement removes traditional functions of buyers, sellers, and intermediaries. The article needs the necessary infrastructure, and we need financial institutions and technology companies to mitigate the risks around building and operating systems that are safe and reliable for non-human actors. Clearly, new protocols for the design and management of authenticating, dispute management, and agent regulation will need to accommodate a different kind of user than we originally designed. It remains to be seen what financial policies and processes can be adapted for these new transactions, since all of the existing policies were designed with people in mind. While the amalgamation of artificial intelligence (AI) with financial technology offers operational efficiencies and can scale quickly, it can also produce ethical and security risks that will require appropriate oversight. Transparency about each software agent's level of controls, accountability, and human agency is integral to consumer acceptance. Moreover, the automaticness of algorithmic decisions moves unique value-based considerations that have traditionally been performed by individuals into solely ethical dilemmas. Beyond the technical aspects of agent-driven commerce, there are changes to the competitive landscape and labour processes associated with economic participation. If we are ever going to adapt to these new autonomous agents, we will need balanced solutions to encourage innovation while regulating toward social, ethical, and regulatory considerations that ensure the gains of new economies are harnessed to benefit all people equitably.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.

- **Funding information:** The authors declare that there is no funding to be acknowledged.

- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] David M. Rothschild, et al., "The Agentic Economy.," arXiv:2505.15799, May 29, 2025. https://arxiv.org/abs/2505.15799

[2] Panos E. Kourouthanassis and George M. Giaglis. "Introduction to the Special Issue: Mobile Commerce – The Past, Present, and Future of Mobile Commerce Research." International Journal of Electronic Commerce, Vol. 16, No. 4, Summer 2012. https://www.jstor.org/stable/41739747

[3] Wasim Rajput. "E-Commerce Systems: Architecture, Design, and Implementation." Artech House / IEEE Xplore, 2000. IEEE Xplore Digital Library. https://ieeexplore.ieee.org/book/9100856

[4] Anan Jin, et al., "DeCoAgent: Large Language Model Empowered Decentralized Autonomous Collaboration Agents Based on Smart Contracts." IEEE Access, October 16, 2024. DOI: 10.1109/ACCESS.2024.3481641. https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?arnumber=10720018

[5] Deli Yang, et al. "Mobile Payment Pattern Based on Multiple Trusted Platforms – China Case." Proceedings of the 2010 Ninth International Conference on Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR), June 24, 2010. https://ieeexplore.ieee.org/document/5494847

[6] G.A.S. Torrellas, et al., "An Authentication Protocol for Agent Platform Security Manager." EFTA 2003 – IEEE Conference on Emerging Technologies and Factory Automation, November 24, 2003. https://ieeexplore.ieee.org/abstract/document/1247764/authors#authors

[7] IEEE Standards Association. "The IEEE Trusted Data & Artificial Intelligence Systems (AIS) Playbook for Finance Initiative." June 4, 2020. Date Added: May 7, 2020 (ICAID Version 1.0). https://standards.ieee.org/wp-content/uploads/import/governance/iccom/IC20-008-Trusted_Data_AIS_Finance.pdf

[8] Bikash Saha, et al. "Generative AI in Financial Institution: A Global Survey of Opportunities, Threats, and Regulation," arXiv:2504.21574, April 30, 2025. https://arxiv.org/abs/2504.21574

[9] Muhammad Hataba, et al. "Security and Privacy Issues in Autonomous Vehicles: A Layer-Based Survey." IEEE Access, 25 April 2022. https://ieeexplore.ieee.org/document/9762777

[10] Leila Ismail, "Authentication Mechanisms for Mobile Agents," Proceedings of the Second International Conference on Availability, Reliability and Security (ARES'07), April 23, 2007. https://ieeexplore.ieee.org/document/4159810

[11] Sharon Goldman, "Why Anthropic's MCP is poised to revolutionize AI-driven e-commerce," LinkedIn, accessed [date]. https://www.linkedin.com/posts/sharongoldman_why-anthropics-mcp-is-poised-to-revolutionize-activity-7328855747977515008-CLxQ

[12] "Announcing the Agent2Agent Protocol (A2A) - A New Era of Agent Interoperability," Google Developers Blog, accessed [date]. https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/