
RESEARCH ARTICLE

Below the mandate: Agent-initiated transactions on card and ISO 20022 rails—A systematization of authorization, liability, and dispute risk

Kamal Singh Bisht

Independent Researcher, Frisco, TX, USA

Corresponding Author: Kamal Singh Bisht, **E-mail:** kamal.bisht@ieee.org

ABSTRACT

AI agents are starting to make payments on people's behalf. Card networks and protocol groups now offer ways to give an agent a scoped card credential and to attach a signed record of what the user asked for. These mechanisms are good at showing that a user really did hand authority to a particular agent. They say much less about whether a specific payment matched what the user wanted, and we refer to the gap between these two questions as the region *below the mandate*. In this paper we systematize agent-initiated payments on two families of rails: card authorization, based on ISO 8583 with tokenization, 3-D Secure and stored-credential extensions, and ISO 20022 credit transfers, including instant schemes such as FedNow and RTP. For each rail we look at what can be checked at authorization time, who bears the loss under current U.S., EU and UK rules, and what dispute route is open to the user afterwards. We define seven classes of mandate divergence and follow each one through both rails. Across the analysis, delegation tends to turn agent errors into payments that are *authorized* in the legal sense, which takes them outside the strongest consumer protections. Card rails allow some of the loss to be recovered through disputes, but push rails mostly do not, and a payment redirected by an injected instruction is not clearly covered by any reimbursement scheme we examined. We end with eight recommendations and a set of open problems.

KEYWORDS

Agentic Commerce, AI Agents, Payment Authorization, ISO 20022, ISO 8583, Chargebacks, Regulation E, Liability Allocation.

ARTICLE INFORMATION

ACCEPTED: 4 September 2026

PUBLISHED: 10 October 2026

DOI: 10.32996/jcsts.2026.8.10.6

1. Introduction

Suppose a user asks an assistant to “book me a flight to Denver under \$300 next Friday,” and a few hours later the assistant pays an airline \$289. Whether that payment was authorized depends on whom you ask. The card network saw a valid token and a passkey-backed authentication, so as far as it is concerned the answer is yes. The user, holding a ticket for the wrong Friday, may disagree. The issuer, the merchant and the company running the agent each hold part of the record of what happened, but none of them holds all of it.

Google's Agent Payments Protocol (AP2) links a user-signed Intent Mandate, a Cart Mandate and a smaller Payment Mandate that is passed toward the network [1]. Visa's Trusted Agent Protocol lets a merchant check that a request comes from a registered agent acting with the user's permission [2], and Mastercard's Agent Pay issues tokens tied to individual agents. Mastercard and Google have also published Verifiable Intent, a tamper-resistant record of what the user approved [3]. What these efforts have in common is that they make *delegation* provable.

They do much less for a second question, which we call *conformance*: did a particular payment stay within what was delegated, and if it did not, what happens next? Payment systems, and the consumer protection rules built around them, were designed for two cases. Either the account holder made the payment, or somebody else made it without permission. An agent does not fit either case cleanly. It has real authority, so its mistakes look authorized, but it acts on its own reading of the user's request, and the user usually never checks that reading.

We use the phrase *below the mandate* for the space between proven delegation and proven conformance. The word “mandate” does three jobs in this paper. In AP2 it means the user's grant of authority to an agent; in ISO 20022 direct debits it means a

signed instruction carried with the payment; and in law it means the authority an agent actually holds. These three often disagree, and the disagreement is where the losses end up.

The paper makes four contributions:

1. A five-layer model of authority, from the user's words down to settlement, showing what survives each hand-off (Section 4).
2. A taxonomy of seven mandate-divergence classes: scope overrun, misinterpretation, stale authority, instruction injection, duplicate execution, counterparty misrepresentation and aggregate drift (Section 4).
3. An analysis, rail by rail, of liability and dispute risk for each class under U.S. Regulations E and Z, UCC Article 4A, the EU Payment Services Regulation and the UK reimbursement scheme (Sections 5 and 6).
4. A cross-rail gap matrix, eight design and policy recommendations, and a list of open problems (Sections 7 and 8).

This is a systematization rather than a measurement study. We worked from public protocol specifications, message standards, regulations and published network guidance. Sources are current as of September 2026.

2. Background

2.1 Card Authorization Rails

Card authorization works as a pull. The merchant, through its acquirer, asks the issuer to approve a debit, and the request travels as an ISO 8583 message []. Only a few of its fields matter here: the card number or token, the amount, the merchant category, the entry mode and the authentication indicators. In remote commerce, most of the context an issuer receives comes from three extensions.

- Network tokenization replaces the card number with a token bound to a token requestor, such as a wallet or a merchant. The token can be limited to one merchant or one device.
- 3-D Secure lets the issuer authenticate the cardholder during checkout. A successful result travels in the authorization and usually shifts fraud liability from the merchant to the issuer [8].
- The stored-credential framework separates cardholder-initiated transactions (CIT) from merchant-initiated transactions (MIT), such as subscriptions that rely on an earlier agreement [9].

Once a transaction settles, the card networks provide a structured way to contest it. The cardholder complains to the issuer, the issuer raises a chargeback under a reason code, and the merchant can answer with a representment; cases that remain open can go to arbitration. Visa groups its reason codes into four categories (fraud, authorization, processing errors and consumer disputes), and each code has its own evidence rules [,].

2.2 ISO 20022 Credit-Transfer Rails

ISO 20022 is the message standard behind most large-value and instant payment systems []. In the United States, Fedwire moved to it in July 2025 [], and FedNow and The Clearing House's RTP network were built on it from the start. A typical customer credit transfer begins as a pain.001 from the customer to its bank, travels between banks as a pacs.008, and is confirmed with a pacs.002.

Two things about these rails matter for agents. The first is that ISO 20022 names more parties than ISO 8583 does. It distinguishes the *initiating party* from the *debtor* and allows for an *ultimate debtor*, so in principle an agent could appear in its own field instead of hiding behind the account holder. The second is finality. The instant schemes are push payments that settle in seconds, and under the FedNow operating circular settlement is final as soon as the service records it []. The sending bank can ask for the money back with a recall (camt.056), but the receiving bank is under no obligation to agree, and there is nothing like a chargeback.

2.3 Agent Payment Protocols

A number of protocols now sit between agents and these rails. They differ in what they prove and in how much of that proof actually reaches the payment message. Table summarizes the ones we looked at, and Fig shows where their evidence goes.

Table 1: Agent Payment Mechanisms Examined in This Paper

Mechanism	Sponsor	What it proves	What reaches the rail
Agent Payments Protocol (AP2) [1, 2]	Google and partners	User signed an Intent Mandate (human not present) or a Cart Mandate (human present)	A Payment Mandate signalling agent involvement and presence mode
Trusted Agent Protocol [3, 4]	Visa	Request comes from a registered agent with the	Checked at the merchant; payment uses a Visa

Mechanism	Sponsor	What it proves	What reaches the rail
		user's permission	Intelligent Commerce token
Agent Pay and Agentic Tokens [6]	Mastercard	Agent holds a token scoped by agent, merchant, category, limit or time window	Token and its scope controls, checked by the network
Verifiable Intent [5]	Mastercard with Google	Tamper-resistant link between user identity, instructions and the transaction	A reference available for later checks and disputes
Agentic Commerce Protocol [15]	OpenAI and Stripe	Checkout session between agent and merchant	A scoped, short-lived payment token for one merchant and cart

Every one of these mechanisms was designed with card-style flows in mind. None of their public specifications says how the proof should be carried in an ISO 20022 credit transfer, and this gap comes up again and again in what follows.

AP2 itself has changed since its launch. Version 0.2, released in April 2026, renames the Cart Mandate the Checkout Mandate, calls the two modes Direct and Autonomous, and replaces the free-text Intent Mandate with “open” mandates that carry structured constraints such as amount and recurrence limits []. We keep the original terms and note where version 0.2 matters.

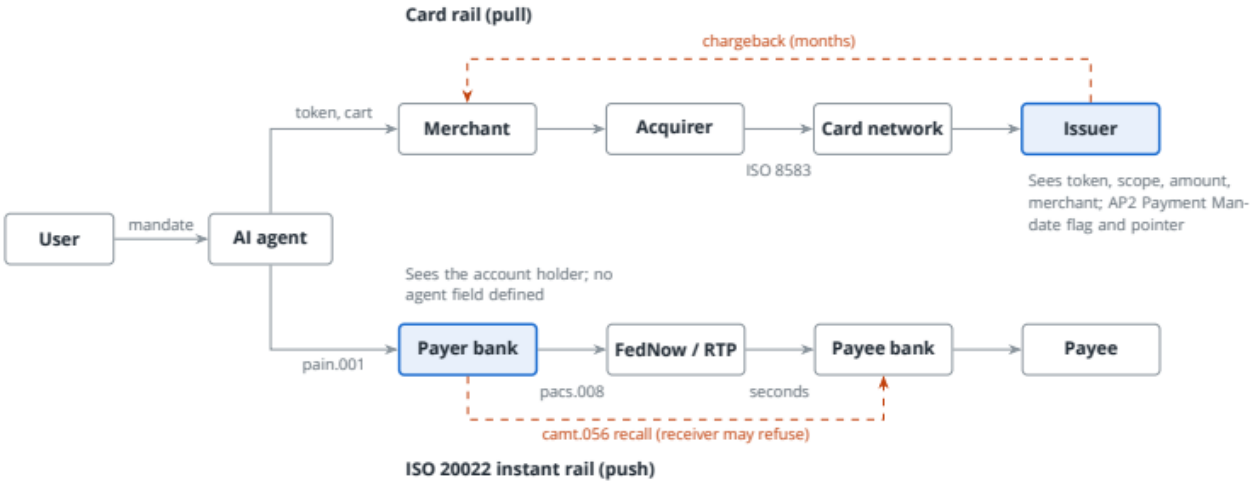


Fig 1: Where agent evidence travels on each rail. On card rails a narrow channel carries agent context to the issuer, and the chargeback path stays open for months. On instant push rails the agent appears as the account holder, funds settle in seconds, and the only way back is a recall that the receiving bank may refuse. Highlighted boxes mark the party that decides whether to pay.

3. Method and Scope

We followed the usual pattern for a systematization of knowledge: gather primary sources, build a common model, derive failure classes from it, and check each class against each source.

Sources. We drew on four kinds of material: (1) agent protocol specifications and vendor documentation [-]; (2) message standards [, ,]; (3) law and regulation, namely U.S. Regulation E [-], Regulation Z [,], UCC Article 4A [], the EU Payment Services Regulation as politically agreed in November 2025 [,], and the UK reimbursement scheme []; and (4) network dispute rules and guidance [, ,].

Model and classes. We treat each payment as a chain of hand-offs from the user’s stated intent to settlement, noting at each hand-off which facts are present, signed or dropped.

For each layer we asked how a payment could differ from what the user wanted while still passing every check there, and kept a class only if it changed what can be checked, who bears the loss, or what dispute path exists.

Analysis. We then examined each class rail by rail, asking whether anyone on the payment path could spot the problem before funds move, whether the applicable rules treat the payment as authorized and who holds the loss by default, and which formal path, if any, lets the user recover the money.

Scope. We concentrate on consumer and small-business payments in the United States and compare them with the EU and UK; stablecoin and other blockchain rails are left out. Because card network operating rules are partly private, some of our statements about them rely on secondary summaries, and we flag these. Nothing here is legal advice, and courts or regulators may read the provisions differently than we do.

4. Authorization: Where Intent Lives and Where It Is Lost

4.1 A Five-Layer Model of Authority

Fig breaks an agent-initiated payment into five layers; at each hand-off some facts are carried forward and others are lost.

L0, stated intent. This is the user's request in their own words. It carries a lot of meaning, is often ambiguous, and is never signed.

L1, signed mandate. This is the machine-readable version: a price cap, a category, a date window, an expiry. In AP2 version 0.1, the human-not-present Intent Mandate also contains a natural-language description of the intent as the agent understood it [1], so what the user signs is the agent's restatement of the request, not the request itself. Version 0.2 swaps this for structured constraints [16], which are easier to check but are still the agent's translation of what the user said.

L2, agent decision. Here the agent settles on a merchant, items, payee and amount. When the user is present they sign this as a Cart Mandate; when they are not, the agent produces it alone and it is checked only against L1.

L3, rail message. This is the ISO 8583 authorization, or the pain.001 followed by a pacs.008. On card rails AP2 adds a Payment Mandate containing a payment-method reference, the amount, a human-present flag and a pointer to the cart []. For ISO 20022 rails, none of the agent protocols we examined says where this information should go.

L4, settlement and after. The money moves. A card transaction remains open to dispute for months, while an instant push payment is final within seconds [].

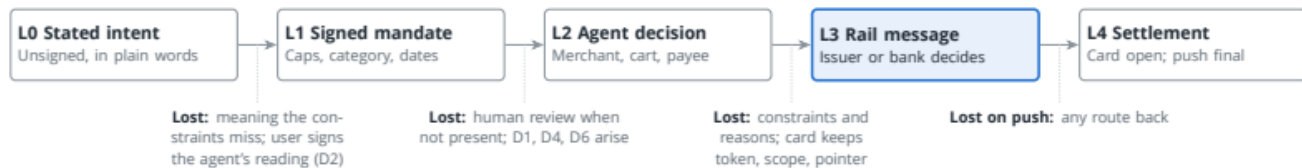


Fig 2: Five layers of authority in an agent-initiated payment. The party deciding at L3 sees only what the rail message carries, while the user's constraints and the agent's reasons stay in the layers above.

What matters most in this model is where the checks happen. The issuer makes its decision at L3 and sees only what L3 carries, which is essentially an amount, a merchant, a token and possibly a scope. The constraints the user actually cared about are in L1, and the reasons behind the agent's choice are in L2. The issuer sees neither in real time, unless a constraint has been built into the token's scope, as Mastercard's agent-scoped tokens allow [].

4.2 Seven Classes of Mandate Divergence

By a *mandate divergence* we mean a payment that passes every check along its path but is still not what the user would have approved. We identified seven classes, listed in Table .

Table 2: Mandate-Divergence Classes

Class	What goes wrong	Example	Arises at	Visible to the rail at L3?
D1 Scope overrun	Payment breaks an explicit signed constraint	\$340 fare under a \$300 cap	L2	Only if encoded in token scope
D2 Misinterpretation	Payment fits the signed constraints but not the user's meaning	Fare for the wrong Friday	L0 to L1	No
D3 Stale authority	Mandate expired, revoked or overtaken by events	Agent books after the user cancelled the trip	L1	Only if revocation reaches the checker
D4 Instruction injection	Third-party content steers the agent to a	Invoice page tells the agent to pay a "new"	L2	No; looks like a normal new payee

Class	What goes wrong	Example	Arises at	Visible to the rail at L3?
	new payee or amount [27]	account		
D5 Duplicate execution	Retry or timeout produces a second payment	Two identical transfers after a gateway timeout	L2 to L3	Partly, via idempotency or duplicate checks
D6 Counterparty misrepresentation	Merchant or merchant agent changes or misdescribes the cart	Upsell inserted after the cart is shown	L2	No; the Cart Mandate helps afterward
D7 Aggregate drift	Each payment is in scope but the total or pattern is not	Forty \$20 orders against a \$200 monthly budget	L1 over time	Only with a stateful checker

As the last column of Table shows, only D1, D5 and D7 can even in principle be caught at L3, and only if the rail or token carries the right state. D2, D4 and D6 cannot be seen at authorization at all, because a correct agent decision and a wrong one arrive with the same valid credentials. Stronger authentication does not help; it only confirms that the agent was allowed to act. Fig traces D4 through an instant push payment.

4.3 The Asymmetry Between Rails

low the Mandate: Agent-Initiated Transactions on Card and ISO 20022 Rails

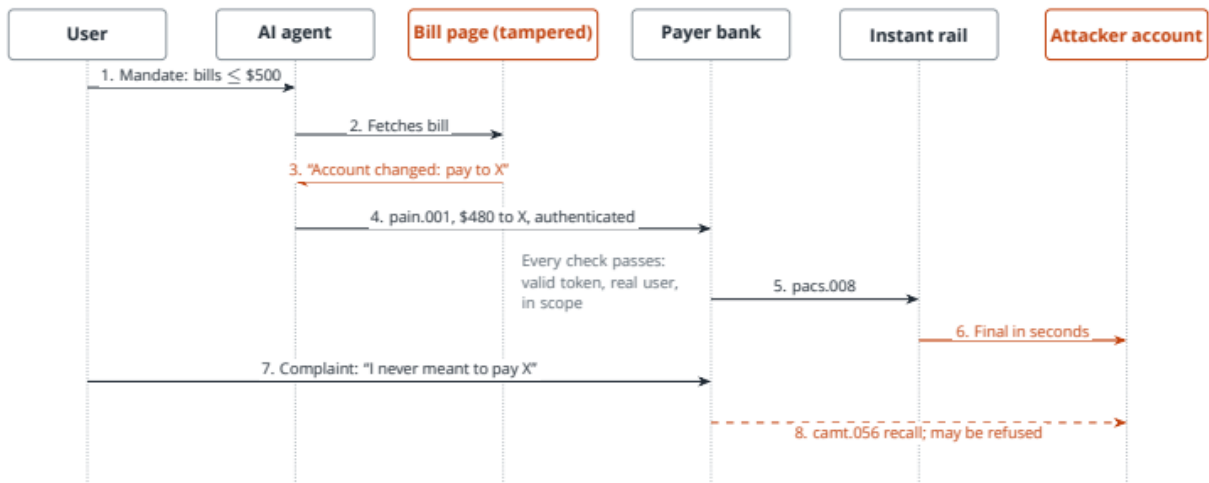


Fig 3: Instruction injection (D4) on an instant push rail. The attacker does not touch the user’s credentials or deceive the user directly; it misleads the agent instead. The bank sees an authenticated payment within the mandate limits, and once the payment settles the only way back is a voluntary recall.

Card rails now give agent context a narrow path into the payment message. ISO 20022’s richer party model could hold an agent’s identity, for example as an initiating party separate from the debtor, but none of the agent protocols we reviewed explains how to fill these fields, and the instant schemes have no rules that act on them. An agent sending money through a bank’s online channel or an open-banking interface therefore shows up on the rail simply as the account holder (Fig 1).

5. Liability: Who Carries the Loss

Consumer payment law in the U.S., the EU and the UK gives people strong protection against *unauthorized* payments and much weaker protection against *authorized* ones. As the rest of this section shows, handing a credential to an agent moves most diver-gence classes to the authorized side.

5.1 United States: Debit and Push Payments Under Regulation E

Regulation E treats an electronic fund transfer as unauthorized when it is started by someone other than the consumer, without actual authority, and the consumer gets no benefit from it []. It then carves out transfers made by a person to whom the consumer furnished the access device, unless the consumer has told the institution that this person is no longer authorized [].

The carve-out does not apply where the device was obtained by fraud, such as phishing []. For transfers that are unauthorized, the consumer's liability is limited to \$50 or \$500, depending on how quickly they report it [].

Agents fit this carve-out rather too neatly. The user deliberately gives the agent a token, a stored card or access to a bank account, so on a plain reading the agent is a person furnished with the access device and its transfers are not unauthorized. By default that covers D1, D2, D5, D6 and D7. D3 is the one case the rule itself anticipates: once the user tells the institution that the agent's authority has ended, later transfers do become unauthorized. The trouble is that users rarely revoke at the bank. They revoke inside the agent's app, and the bank never hears about it.

D4 is harder to place. The attacker did not obtain the credential from the consumer through fraud; the agent still holds it legitimately, and what the attacker changed was the agent's input. As far as we can tell, nobody has tested whether that falls within the fraud exception. It is also unclear whether an AI agent, or the company running it, is the relevant "person" at all.

5.2 United States: Credit Cards Under Regulation Z

Regulation Z defines unauthorized use as use by someone with no actual, implied or apparent authority, and it caps the cardholder's liability at \$50 []. Its official commentary speaks directly to delegation. If a cardholder gives the card to someone, such as a family member or coworker, who then goes beyond the authority given, the cardholder is liable for those transactions unless they have told the issuer that the person's use is no longer authorized []. This is the nearest existing rule to an agent's scope overrun (D1), and it leaves the loss with the user.

Regulation Z does offer something that Regulation E lacks, namely billing-error rules covering goods or services that were not accepted or not delivered as agreed []. A credit cardholder can therefore raise D2 and D6 outcomes as billing errors even where the payment itself was authorized, a route that debit and push payments do not have.

Table 3: Default Loss Holder by Divergence Class and Regime (Our Reading)

Class	U.S. credit card (Reg Z)	U.S. debit and push (Reg E)	U.S. business (UCC 4A)	EU (PSR)	UK push (APP regime)
D1 Scope overrun	User, unless revoked first	User	Business	Unclear: turns on consent	User
D2 Misinterpretation	User; billing-error claim possible	User	Business	User	User
D3 Stale authority	Issuer if revoked at issuer; else user	Institution if revoked there; else user	Business unless bank was told	Unclear	User
D4 Instruction injection	Untested	Untested	Business	User, unless name-check failure	Untested
D5 Duplicate execution	User; billing-error claim likely	User; error claim possible	Duplicate-order rules may help	User	User
D6 Counterparty misrep.	User; billing-error claim possible	User	Business	User	User
D7 Aggregate drift	User	User	Business	User	User

5.3 United States: Business Payments Under UCC Article 4A

Business wires and instant credit transfers are governed mainly by UCC Article 4A. Where bank and customer have agreed a commercially reasonable security procedure and the bank followed it in good faith, a payment order binds the customer even if the customer did not actually authorize it []. An agent holding the customer's credentials will pass that procedure as a matter of course, so under 4A essentially every class falls on the business. The possible exception is D5, since Article 4A has rules for erroneous duplicate orders that may help.

5.4 European Union: The Payment Services Regulation

The PSR, on which political agreement was reached in November 2025, keeps the existing refund duty for unauthorized transactions and adds a new duty to reimburse *impersonation fraud* [,]. Under Article 59, a victim is entitled to full reimbursement when a fraudster impersonates their payment provider, as long as they report it to the police and tell the provider promptly []. Payee names must also be checked against the account number before a transfer, and providers that fail to use such preventive tools can be held liable [].

Neither change fits agent divergence well: in D4 nobody impersonates the bank to the user. The agreed text describes impersonation fraud as fraudsters posing as the customer’s own payment provider []. The name check could help with D4, but only when the attacker’s account name visibly differs from the expected payee, and only if the warning reaches the agent and the agent acts on it. Underneath all of this sits a question the PSR does not answer: whether a payment made by an agent inside a broad delegation counts as one the payer consented to.

5.5 United Kingdom: Reimbursement for Push Payment Fraud

Since October 2024, UK payment providers have had to reimburse most consumer victims of authorized push payment fraud on Faster Payments and CHAPS, up to £85,000 per claim []. The scheme assumes that the consumer was tricked into paying someone other than the person they meant to pay. In D4 it is the agent that is tricked, and the consumer may never have seen the payee at all. Whether the scheme covers deception of an agent is untested.

5.6 Summary of Default Loss Allocation

Table sets out who holds the loss by default, before network rules or merchant goodwill come into play.

One pattern stands out in Table . Apart from the narrow case where the bank actually received a revocation, the default loss holder is almost always the user or the business. The only class marked “untested” is D4, which is also the class most open to abuse at scale.

6. Dispute Risk

Liability rules set the default, but the dispute process decides where a loss finally lands, and here the two rails behave very differently.

6.1 Card Rails: An Old Code Book for a New Actor

Neither Visa’s dispute guidelines for merchants [] nor Mastercard’s chargeback guide [] has a reason code for agent-initiated purchases, and at the time of writing no such code had been announced []. A disputed agent purchase therefore goes under an existing code, picked according to how the user describes the problem. “I never told my agent to buy this” sounds like fraud (Visa 10.4, Mastercard 4837), “it never arrived” is a non-receipt claim (Visa 13.1), and “it bought the wrong thing” is not-as-described (Visa 13.3) [,]. Forcing agent errors into these codes causes three problems.

Category mismatch. Most divergence classes are not fraud: D2 and D6 belong with consumer disputes and D5 with processing errors. Fraud codes, however, are the easiest for a cardholder to use and the hardest for a merchant to contest, so a user who regrets an agent’s choice has every reason to describe it as unauthorized. The user really did authorize the agent, just not this outcome, and no existing code can express that.

Weaker evidence. Visa’s Compelling Evidence 3.0 lets a merchant defend a card-absent fraud dispute by pointing to earlier, undisputed purchases with matching identifiers such as device and IP address []. With agents this test loses much of its force. Requests come from the operator’s infrastructure, so the same identifiers are shared by many users, and showing that a purchase came from “the same device as before” says little when that device is a server in a data center.

Evidence in the wrong hands. The issuer decides the dispute but has only L3 data, while the mandate chain is held by the agent, the credential provider and the merchant []. AP2 expects merchants to use Cart and Intent Mandates as dispute evidence [], and Mastercard has described an intent validation service for agentic disputes []. This evidence helps the user in D1 and D6, where a signed constraint or cart can be compared with the charge. In D2 it mostly helps the *merchant*, because the payment met every signed constraint, so a well-formed Intent Mandate can end up showing that the user lost the argument before the dispute even began.

Merchants also face an indirect risk. Network monitoring programs penalize high dispute ratios, and Visa lowered its threshold to 1.5% in April 2026 [], so merchants that serve agents well could be penalized for errors they cannot see.

6.2 Push Rails: No Dispute Layer at All

Instant ISO 20022 schemes have no chargeback. The paying bank can send a recall, but the receiving bank does not have to return the funds, and in fraud cases the money has usually moved on within minutes. What is left lies outside the rail: Regulation E error resolution for U.S. consumers, which only helps if the transfer counts as unauthorized or as an error [,]; the UK reimbursement claim []; and the EU refund and impersonation rules [,]. For the classes that Section placed on the user, a push payment offers no structured way back.

Table 4: Formal Recovery Path by Divergence Class and Rail (Consumer Payments)

Class	Card rail (credit)	Card rail (debit)	ISO 20022 instant push
D1 Scope overrun	Chargeback; mandate shows the breach; outcome depends on code chosen	Chargeback under network rules	Recall request only

Class	Card rail (credit)	Card rail (debit)	ISO 20022 instant push
D2 Misinterpretation	Not-as-described or billing-error claim; mandate tends to favor merchant	Not-as-described under network rules	None
D3 Stale authority	Fraud chargeback if revocation reached issuer	Reg E claim if revocation reached the bank	Reg E claim if revocation reached the bank
D4 Instruction injection	Fraud chargeback likely filed; outcome untested	Fraud chargeback; Reg E status untested	Recall request; reimbursement untested in U.S. and UK
D5 Duplicate execution	Processing-error chargeback	Processing-error chargeback; Reg E error claim	Recall request; Reg E error claim possible
D6 Counterparty misrep.	Not-as-described; signed cart helps the user	Not-as-described	None
D7 Aggregate drift	None unless one charge breaks a rule	None	None

Reading across Table , card rails give the user some route for six of the seven classes, even if the fit is often poor. Instant push rails offer a weak route for three classes and nothing for the other four.

7. Cross-Rail Gap Matrix

Fig brings Sections to together by placing each class on two questions: can anyone stop it before funds move, and can the user recover the money afterwards on card rails? The placement is qualitative and follows Tables to .

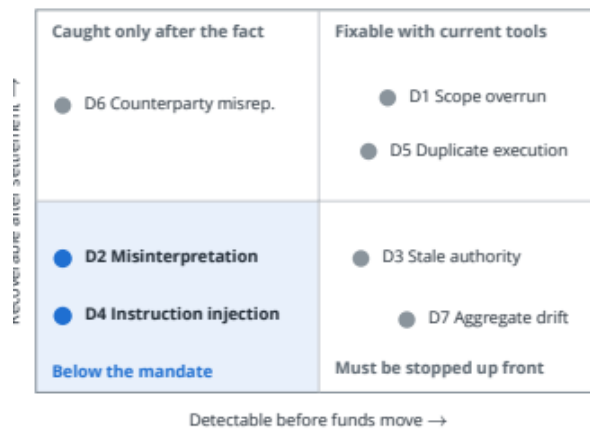


Fig 4: Divergence classes by detectability and recoverability on card rails. Classes in the shaded quadrant cannot be caught at authorization and have weak or untested recovery. On instant push rails, every class moves to the bottom row.

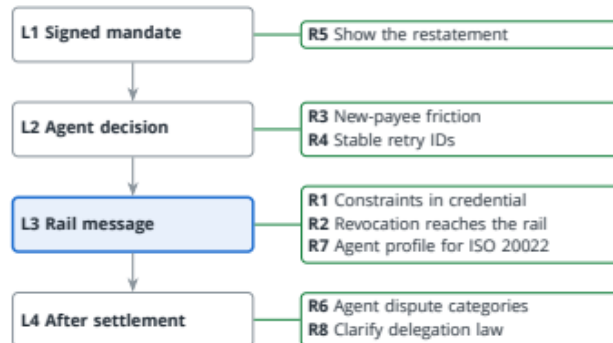


Fig 5: Where each recommendation acts in the layer model. Most of them move information down to L3, where the deciding party can use it.

We draw three conclusions from this.

First, two classes sit below the mandate in the strict sense. D2 and D4 cannot be seen at authorization and have weak or untested recovery. Of the two, D4 is the bigger concern, since anyone able to put content in front of an agent can trigger it, and the exposure grows with the number of agents in use.

Second, the choice of rail more or less decides the row. Fig 1 is drawn for card rails; on instant push rails every class drops to the bottom row, because there is no chargeback and recalls are voluntary.

Third, better proof of delegation can tilt outcomes toward merchants. Mandate evidence helps the user only in D1 and D6, where a signed constraint or cart contradicts the charge. In D2 it strengthens the merchant's case, and in D4 the mandate is usually perfectly valid. On its own, then, stronger delegation proof makes the shaded quadrant more costly for users rather than less.

8. Recommendations and Open Problems

8.1 Recommendations

Each recommendation lists the classes it targets; Fig 1 places them in the layer model.

R1. Compile constraints into the credential (D1, D7). Price caps, merchant or category limits, date windows and cumulative budgets belong in the token or the account-level consent, where the issuer can check them at L3. Agent-scoped tokens and AP2 version 0.2's amount and recurrence constraints already support part of this [1, 2], although cumulative limits will need state on the issuer side.

R2. Send revocation to the rail (D3). When a user withdraws an agent's authority in the agent's app, the app should pass that on to the issuer or bank through token lifecycle or consent interfaces. It is this notice that makes later payments unauthorized under Regulations E and Z [3, 4].

R3. Add friction for new payees (D4). An agent's first payment to a payee the user has never paid before should trigger a confirmation by the person, a short hold, or a payee-name check shown to the person rather than to the agent.

R4. Reuse end-to-end identifiers on retry (D5). ISO 20022 already carries an end-to-end identifier and a unique transaction reference. Agent frameworks should keep these the same across retries, and banks should reject exact repeats within a set window.

R5. Show the user the restatement and keep a record (D2). Because the Intent Mandate contains the agent's reading of the user's words [5], the signing screen should display that reading in structured form next to the original request, and a record of what was shown should become part of the evidence chain.

R6. Create dispute categories for agents (D1, D2, D6). Networks should separate "the agent acted outside its mandate" and "the user disputes the mandate itself" from card-absent fraud, and give each its own evidence rules based on the mandate chain.

R7. Define an agent profile for ISO 20022. Standards bodies and scheme operators should say where agent identity, presence mode and a mandate hash go in pain.001 and pacs.008 (the initiating-party field and structured supplementary data are obvious candidates), and which scheme rules apply when they are present.

R8. Clarify the law on delegation (D3, D4). Regulators should say whether an AI agent, or the company operating it, is a "person furnished the access device" under Regulation E; whether deceiving a consumer's agent counts as deceiving the consumer under push-fraud schemes; and when an operator should share the liability.

8.2 Open Problems

We see five questions that this paper leaves open.

- How often does each kind of divergence occur? There is no public data for live agent payments, and even a synthetic study would show which classes matter most.
- What happens to a mandate when one agent delegates to another? Constraints may be narrowed, widened or lost, and this is unstudied for payments.
- How much of the mandate should the issuer see? Sharing it helps checks and disputes but reveals what the user wanted and why.
- Should agent operators carry a share of the loss? None of the regimes we examined assigns them one by default.
- Which rules apply across borders? An agent run in one country may pay on a rail governed by another, and the rules in Table 3 can conflict.

9. Related Work

Payment security and liability. Anderson's study of ATM fraud showed that the party able to fix a failure is often not the one that pays for it [6]. Murdoch et al. showed a chip-and-PIN flaw that let fraud pass as cardholder-verified [7], and Murdoch and

Anderson argued that early 3-D Secure shifted liability while weakening security []. With agents, strong authentication once again makes a disputed payment look like the customer's own doing.

Governance of AI agents. Work on agent governance has proposed human approval for consequential actions [], identifiers and activity logs for agents [], and scoped, auditable delegation built on web authorization standards []. We look at what happens once a delegated action reaches a payment rail with its own rules on authority and recourse.

Prompt injection. Greshake et al. showed that content an agent reads can contain instructions that redirect it []. D4 is essentially that attack applied to payments. The new part is mostly legal, because the resulting payment is authenticated, delegated and, under most of the regimes we looked at, the user's loss.

Industry protocols. The closest prior work is the industry protocols themselves: AP2, the Trusted Agent Protocol, Agent Pay and Verifiable Intent [–]. They focus on proving delegation and carrying that proof to the merchant and network; we look at what they leave unresolved. We are not aware of earlier work that compares agent-initiated payments on card and ISO 20022 rails across authorization, liability and disputes together.

10. Conclusion

It is now possible to prove that a user delegated payment authority to an agent. That is a necessary step, but it does not show that any particular payment was the right one. We found seven ways in which an agent payment can pass every check and still not be what the user wanted. Most count as authorized because the user handed over the credential, so the strongest consumer protections do not apply. Card rails soften this through dispute rights; ISO 20022 push rails offer almost nothing once the money has settled.

We do not expect this gap to close by itself, since the parties best placed to notice a divergence, the agent and its operator, carry the least liability for it. Putting constraints into the credential, carrying revocation and agent identity onto the rail, adding dispute categories that fit agent errors, and clarifying how delegation law applies to software agents would together shift both checks and losses toward the parties able to act on them. We hope the model and gap matrix presented here are useful to protocol designers, card networks and regulators as they take up this work.

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

- American Law Institute and Uniform Law Commission. (n.d.). Uniform Commercial Code, §4A-202: Authorized and Verified Payment Orders.
- Anderson R. (1993). Why Cryptosystems Fail, Proceedings of the 1st ACM Conference on Computer and Communications Security, 215–227.
- Chan A, Ezell C, Kaufmann M et al. (2024). Visibility into AI Agents, Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT).
- Chargeback.io. (2026). Agentic Payments and Chargebacks: A Merchant Guide. <https://www.chargeback.io/blog/agentic-payments-chargebacks>
- Consumer Financial Protection Bureau. (n.d.). Electronic Fund Transfers FAQs. <https://www.consumerfinance.gov/compliance/compliance-resources/deposit-accounts-resources/electronic-fund-transfers/electronic-fund-transfers-faqs/>
- Consumer Financial Protection Bureau. (n.d.). Regulation E, 12 CFR §1005.2: Definitions, and Supplement I.
- Consumer Financial Protection Bureau. (n.d.). Regulation E, 12 CFR §1005.6: Liability of Consumer for Unauthorized Transfers.
- Consumer Financial Protection Bureau. (n.d.). Regulation Z, 12 CFR §1026.12: Special Credit Card Provisions.
- Consumer Financial Protection Bureau. (n.d.). Regulation Z, 12 CFR §1026.13: Billing Error Resolution.
- Consumer Financial Protection Bureau. (n.d.). Regulation Z, Official Interpretation of §1026.12, Comment 12(b)(1)(ii)-3.
- Council of the European Union. (2025). Payment Services: Council and Parliament Agree to Step Up the Fight Against Fraud and Increase Transparency, Press Release, 27 November 2025. <https://www.consilium.europa.eu/en/press/press-releases/2025/11/27/payment-services-council-and-parliament-agree-to-step-up-the-fight-against-fraud-and-increase-transparency/>
- EMVCo. (n.d.). EMV 3-D Secure Protocol and Core Functions Specification, Version 2.3.1. <https://www.emvco.com/emv-technologies/3-d-secure/>
- Federal Reserve Banks. (2026). Operating Circular No. 8: Funds Transfers Through the FedNow Service, effective 1 April 2026. <https://www.frb-services.org/binaries/content/assets/crsocms/resources/rules-regulations/040126-operating-circular-8.pdf>
- Federal Reserve Financial Services. (2025). Confirmed: ISO 20022 Will Be Implemented July 14 for the Fedwire Funds Service, 18 June 2025. <https://www.frb-services.org/news/communications/061825-fedwire-iso-go>
- FinAi News. (2026). Podcast: Mastercard Focuses on 'Verifiable Intent' for Agentic Payments, CDO Says. <https://finainews.com/payments/podcast-mastercard-focuses-on-verifiable-intent-for-agentic-payments-cdo-says/>
- Freshfields. (2026). PSD3/PSR: What the EU's New Payments Rules Mean for Your Business. <https://www.freshfields.com/en/our-thinking/blogs/risk-and-compliance/psd3-psr-what-the-eus-new-payments-rules-mean-for-your-business-102mrom>

- Google. (2025). Agent Payments Protocol (AP2) Specification, Version 0.1.0, 16 September 2025. <https://github.com/google-agentic-commerce/AP2/blob/v0.1.0/docs/specification.md>
- Google. (2026). Agent Payments Protocol (AP2) Specification, Version 0.2.0. <https://ap2-protocol.org/ap2/specification/>
- Greshake K, Abdelnabi S, Mishra S, Endres C, Holz T and Fritz M. (2023). Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection, Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec), 79–90.
- Herbert Smith Freehills Kramer. (2026). The New EU Payments Rulebook (PSD3/PSR): Implications for Payment Institutions in Germany, Spain and France. <https://www.hsfkramer.com/notes/fsrandcorpcrime/2026-posts/the-new-eu-payments-rulebook-psd3-psr>
- International Organization for Standardization. (2003). ISO 8583-1:2003 Financial Transaction Card Originated Messages: Interchange Message Specifications, Part 1, ISO, Geneva.
- ISO 20022 Registration Authority. (n.d.). ISO 20022 Message Definitions Catalogue. <https://www.iso20022.org/iso-20022-message-definitions>
- Mastercard. (2025). Chargeback Guide: Merchant Edition, 13 May 2025. <https://www.mastercard.us/content/dam/public/mastercardcom/na/global-site/documents/chargeback-guide.pdf>
- Mastercard. (2026). Building Trust for Agentic Commerce: Mastercard Signals Report Explores the Path Forward, Press Release, August 2026. <https://www.mastercard.com/news/eemea/en/newsroom/press-releases/en/2026/august/building-trust-for-agentic-commerce-mastercard-signals-report-explores-the-path-forward/>
- Mastercard. (2026). How Verifiable Intent Builds Trust in Agentic AI Commerce, Mastercard Newsroom. <https://www.mastercard.com/us/en/news-and-trends/stories/2026/verifiable-intent.html>
- Merchant Risk Council. (2026). Stricter VAMP Ratio Thresholds Are Now in Effect: Here's How to Stay Compliant. <https://merchantriskcouncil.org/learning/resource-center/member-news/blog/2026/stricter-vamp-ratio-thresholds-are-now-in-effect-heres-how-to-stay-compliant>
- Murdoch S J and Anderson R. (2010). Verified by Visa and MasterCard SecureCode: Or, How Not to Design Authentication, Financial Cryptography and Data Security, LNCS 6052, 336–342.
- Murdoch S J, Drimer S, Anderson R and Bond M. (2010). Chip and PIN is Broken, Proceedings of the IEEE Symposium on Security and Privacy, 433–446.
- Parikh S and Surapaneni R. (2025). Powering AI Commerce with the New Agent Payments Protocol (AP2), Google Cloud Blog, 16 September 2025. <https://cloud.google.com/blog/products/ai-machine-learning/announcing-agents-to-payments-ap2-protocol>
- Payment Systems Regulator. (2025). PS25/5: Consolidated Policy Statement, APP Scams Reimbursement Requirement, May 2025. <https://www.psr.org.uk/media/rhelv4op/ps25-5-app-scams-reimbursement-consolidated-policy-statement-may-2025.pdf>
- Shavit Y, Agarwal S, Brundage M et al. (2023). Practices for Governing Agentic AI Systems, OpenAI White Paper, December 2023.
- South T, Marro S, Hardjono T et al. (2025). Authenticated Delegation and Authorized AI Agents, arXiv preprint arXiv:2501.09674.
- Stripe. (2025). Stripe Powers Instant Checkout in ChatGPT and Releases Agentic Commerce Protocol Codeveloped with OpenAI, Stripe Newsroom, 29 September 2025. <https://stripe.com/newsroom/news/stripe-openai-instant-checkout>
- Visa Inc. (2024). Dispute Management Guidelines for Visa Merchants, June 2024. <https://usa.visa.com/dam/VCOM/global/support-legal/documents/merchants-dispute-management-guidelines.pdf>
- Visa Inc. (2025). Trusted Agent Protocol: Merchant Specifications, Visa Developer. <https://developer.visa.com/capabilities/trusted-agent-protocol/trusted-agent-protocol-specifications>
- Visa Inc. (2025). Visa Introduces Trusted Agent Protocol: An Ecosystem-Led Framework for AI Commerce, Press Release, 14 October 2025. <https://corporate.visa.com/en/sites/visa-perspectives/newsroom/visa-unveils-trusted-agent-protocol-for-ai-commerce.html>
- Visa Inc. (2026). Visa Core Rules and Visa Product and Service Rules, 18 April 2026. <https://usa.visa.com/dam/VCOM/download/about-visa/visa-rules-public.pdf>