Agentic AI, Open Finance, & Technical Standards

Exploration Brief

FINANCIAL® DATA EXCHANGE

FINANCIAL® DATA EXCHANGE

1. Introduction

Millions of consumers and business share their financial data with third-party apps and services today. In recent years, broad industry collaboration has enabled a majority of these data sharing connections in North America to transition to the use of structured, token-based APIs aligned with the FDX standards. Those standards are purpose-built to provide enhanced security, user control & transparency, scoped consent management, traceability, and interoperability.

Now, financial institutions, fintechs, AI platform operators, and account holders are increasingly deploying AI agents in a variety of ways to facilitate financial data sharing. These new uses introduce opportunities but also novel risks, while challenging many assumptions imbedded in the technical standards & integration patterns that are widely used today.

This brief provides an initial framing of

- Use cases for agentic AI in permissioned financial data sharing
- How those use cases may introduce new risks or questions; and
- Areas where industry standards may need to adapt

This brief is based off discussions to date in the FDX Agentic AI Task Force, a cross-industry group of financial institutions, data aggregators, fintechs, and others.

After reading this brief, we invite you to review the companion Call for Input document and share your insights and feedback.

2. Scope of inquiry

This Brief is focused on the data-sharing layer where AI agents interact with the user-permissioned financial data ecosystem.

This includes a scenario where agents use consumer credentials to access financial data through (e.g., a bank's) web or mobile interface. It also includes a scenario where AI agents are involved in accessing dedicated third-party API channels.

This Brief examines both the current reality and the emerging patterns – specifically, how intelligent agents (chatbots, autonomous advisors, LLM-powered applications, and similar systems) are or may soon be involved in facilitating consumer- and business-permissioned financial data access.

This inquiry is focused on standards at the app layer, not the model layer. FDX does not intend to interrogate how LLMs work. These models vary widely and evolve rapidly. Instead, our focus is the application layer – which involves the entity holding the API credentials, or that otherwise makes or receives data access requests.

For purposes of this paper, we define an "AI agent" as an application that can take actions (e.g., call APIs, access a website, read/write data, execute workflows) and typically uses an LLM to reason about and determine its next steps.

This brief does not address AI model design or training, broad AI ethics frameworks, legal liability allocation for agent decisions, or baseline agent quality criteria.

Adjacent questions—such as how companies use AI internally to build APIs, detect fraud, or generate insights—are important but fall outside the scope of this inquiry because they do not directly involve how user-permissioned data flows between parties. The regulatory landscape, including unresolved questions about the legal status of AI agents and how existing consent frameworks apply to autonomous behavior, will be monitored by FDX as context for ongoing standards work. [See Endnote 1[i] for additional details.]

2. Examining three roles that Agentic AI plays in user-permissioned financial data sharing

There are three primary roles that an AI agent can play in the permissioned financial data sharing ecosystem. These roles are defined by the agent's relationship to the data flow – not by who operates the agent or what industry it serves. Each role raises distinct governance questions and stress-tests different parts of FDX's existing standards.

(These roles are not rigid categories – a single agent may blur across multiple roles within a single session, and entirely new patterns may emerge that do not fit neatly into this framework. This taxonomy is a starting point for discussion, not a definitive classification of a space that is changing rapidly. Also, "agent" is not monolithic. An agent may be a feature embedded inside an already-registered fintech or banking application—operating under existing agreements and contractual controls—or a general-purpose AI assistant [e.g., operating on a user's device] that a consumer points at their financial account provider's interface, operating outside any existing framework).

Role 1: Agent acting as OAuth API Client

An agent automates the retrieval, and potentially the initiation, of actions through APIs on behalf of an authorized third party, replacing scripted logic hard-coded by developers. The agent holds or acts under the third party's OAuth client credentials.

> ***Illustrative example (read):*** *A fintech company that provides expense categorization currently uses developer-written scripts to pull transaction data from a bank's FDX API nightly. The company replaces those scripts with an AI agent that dynamically determines which accounts to query, what date ranges to pull, and how to handle errors – all operating under the same OAuth credentials the company was previously issued by the bank. From the bank's perspective, the API caller looks identical. But the logic behind the calls is now adaptive and non-deterministic.*
>
> ***Illustrative example (write):*** *A data recipient has access to a bank's payment-initiation API with money movement permissions and deploys an agent to initiate transfers or payments using that third party's API credentials. The agent's ability to take financial actions, not just read data, introduces additional considerations around non-repudiation, per-operation consent, and accountability for actions taken autonomously.*

Questions this could raise (examples): Do data providers need increased visibility into *how* OAuth credentials are being used or if by an agent? Does this new pattern create risks for the consumer or the interacting parties that need to be managed in new ways? When agents can both read data and initiate actions under the same credentials, how should the governance model differentiate between these modes?

Role 2: Agent as Consent Manager

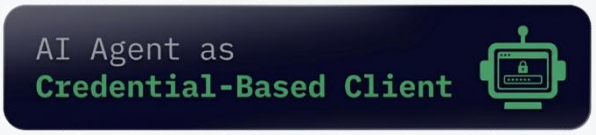
A user grants consent that authorizes an agent to act on their behalf within the permissioned data ecosystem – managing connections, responding to reauthorization requests, and potentially modifying or revoking access over time. The user may (or may not) be on the other side of the initial consent screen. They may authorize an agent to exercise that consent going forward, potentially over extended periods without the user being present. Both patterns - agents that carry forward a user-initiated consent, and agents that independently complete authorization flows - are plausible. The agent may be operated by the fintech application itself (as a built-in feature), by a standalone AI platform the consumer has chosen independently, or by a general-purpose assistant on the consumer's device. Who deploys and operates the agent matters for accountability and varies across implementations.

> ***Illustrative example:*** *A consumer connects their bank account to a fintech budgeting app through the standard OAuth flow, and in doing so grants their AI assistant authority to manage that connection going forward. The connection is established on January 1. Over the next several years, the assistant reauthorizes access each time consent expires – without the consumer being prompted. When the budgeting app requests additional data scopes (say, investment account access for a new tax*

> *optimization feature), the assistant evaluates whether to approve the expanded scope based on its understanding of the consumer's goals, potentially without consulting the consumer in real time.*

Questions this could raise (examples): This use case directly challenges the assumption that a conscious human is on the other side of every consent screen. It raises questions about the duration and boundaries of delegated authority, whether agents can expand consent scope autonomously, and what "meaningful consent" looks like when the party completing parts of the user consent process is software.

Role 3: Agent as Credential-Based Access Client



An agent uses the consumer's login credentials to access financial data through web or mobile interfaces outside a dedicated third-party API channel – i.e., an agentic-era version of "screen-scraping." In this pattern, the agent may (or may not) be controlled by the consumer directly – either through a standalone AI assistant, a locally-run application, or a consumer-facing AI platform. The consumer may be both the authorizing party and the party solely responsible for the agent's behavior (i.e., there is no organizational data recipient to hold accountable if the agent exceeds its authorization), which may complicate traditional accountability models that assume an organizational data recipient.

> ***Illustrative example:*** *A consumer gives their bank username and password to an AI assistant and asks it to "keep track of all my finances." The assistant logs into the bank's web portal, navigates the account dashboard, and extracts transaction data by reading the rendered page. From the bank's perspective, this session may be indistinguishable from a human customer logging in – but the "customer" is software, operating persistently, and potentially accessing the account more frequently and broadly than any human would. In some cases, the "scraping" may be performed by an agent operating on the consumer's device (and not by software run by a company from the cloud).*

Questions this could raise (examples): How should data providers detect and manage credential-based agent traffic that operates outside API channels? What factors would make the API-based path more attractive to agent developers — and is this ultimately a more durable response than detection alone? How can FDX standards evolve to support the adoption of secure data access methods for agentic use cases? What distinguishes the "agentic scraping" pattern from legacy screen-scraping?

Agent-Initiated Actions: Raising the stakes

A variation on all three use cases above occurs where agents cross from reading data to initiating actions at financial institutions (e.g., moving money, paying bills, opening accounts) based on their own reasoning. When agents cross this "actuation boundary," the stakes escalate.

Cross-cutting dimensions

In addition to the three roles above, several dimensions cut across all use cases.

Human supervision	Fully supervised ↔ Fully autonomous
Operator type	Embedded in registered app (e.g., fintech or bank mobile app) ↔ Consumer-directed general-purpose assistant
Access pattern	Read-only retrieval ↔ Write (e.g., payments, account changes)
Persistence	Single session ↔ Persistent, multi-month delegation

The standards implications shift along each of these dimensions. An agent operating autonomously, initiated by a consumer, performing write operations over an extended period represents a fundamentally different governance challenge than a supervised agent embedded in a registered fintech performing read-only queries within a single session.

3. Across these three roles, Agentic AI introduces six new challenges to be solved

3.1 Agent Identification and Classification

Applies primarily to: Use Case 3 (credential-based access). Secondarily: Use Case 1 (API client).

In the API context (Use Case 1), a data provider has issued OAuth client credentials to a known company. Anyone presenting valid credentials is either that company or a tool acting on its behalf. Existing mechanisms (e.g., OAuth client authentication, contractual terms) continue to apply. The identification question in this context is narrower: should a data provider know that an agent (rather than a deterministic, hard-coded script) is making access decisions under those credentials, and does that knowledge change the risk profile or the terms of access?

The identification challenge becomes acute in Use Case 3. When an agent accesses a bank's web portal using a consumer's login credentials, the bank may not distinguish it from a human customer, an authorized agent acting on the customer's behalf, or an unauthorized bot. Most bot-detection systems in production today were not designed to distinguish between unauthorized automation and consumer-authorized agentic delegation.

> ***Illustrative example:*** *A bank observes a spike in sessions that log in, navigate to transaction history, and export data in a consistent pattern. Some of these sessions are consumers using AI assistants they've authorized. Others are unauthorized scrapers. The bank has no standardized mechanism to distinguish between them and risks blocking legitimate, consumer-authorized agents in order to prevent fraud.*

A multi-layered identification challenge. Agent identification is not a single question – it likely requires resolving at least three distinct layers: (1) the agent itself – what application is acting, who operates it, and under what terms; (2) the underlying model or reasoning engine – which LLM or AI system is powering the agent's decision-making; and (3) the delegating consumer (or another agent) – on whose behalf is the agent acting? Current identification mechanisms typically address only one of these layers (if any). A comprehensive approach would require consistent, interoperable signals across all three.

Where else this is evolving. The "Know Your Agent" (KYA) concept is rapidly gaining market traction, with a growing ecosystem of vendors offering agent identification and verification products – a signal that demand for this capability is real and accelerating. However, much of the current market focus is on verifying the agent itself (layer 1 above), with less attention to linking agent identity to the consumer who authorized it (the KYC-attached-to-KYA problem). NIST's NCCoE published a concept paper in February 2026 on accelerating the adoption of software and AI agent identity and authorization standards, focusing on enterprise contexts. Cloudflare is developing mechanisms (e.g., WebBotAuth) to distinguish between types of automated web clients. FDX's contribution could be to extend these emerging patterns to the specific requirements of consumer-permissioned financial data access.

3.2 Delegation and Consent

Applies primarily to: Use Case 2 (consent manager). Secondarily: Use Case 1 (API client).

Current open banking standards assume a human-initiated session with a clear beginning and end. A user logs in, grants consent, and an application accesses data within defined boundaries. Agents introduce autonomous, asynchronous, and persistent access that stress these assumptions.

> ***Illustrative example:*** *A consumer asks an AI assistant to connect their bank account to a fintech app and "keep it connected." The assistant completes the initial OAuth flow. Twelve months later, when reauthorization is required, the assistant approves it without consulting the consumer. Six months after that, the fintech requests access to investment accounts for a new feature; the assistant evaluates the request against its understanding of the consumer's financial goals and approves the expanded scope. The consumer is never prompted. Three years have now passed since the consumer's original instruction.*

The design challenge: how do we build consent frameworks that support meaningful delegation *and* meaningful user control? A static "yes" at the beginning of a multi-year relationship may be insufficient. Yet repeatedly prompting users for every asynchronous action creates consent fatigue that undermines the value of delegation. The identification question becomes layered: who is "acting" – the user, the agent, or the platform operating the agent?

Emerging thinking: Concepts like digitally signed consent, verifiable credentials, step-up authentication for high-risk agent actions, and time-bounded delegation windows are being discussed.

3.3 Data Minimization and Scope Fluidity

Applies to: All three use cases, but the challenge is most pronounced for Use Cases 1 & 2.

Traditional fintech applications have a fixed purpose that a developer designed. A budgeting app requests transaction data; a payment wallet app requests account balances and other account information. The purpose is static, the data needs are predictable, and scopes can be matched to that fixed purpose.

Agent-based applications are different. The very nature of what the application does can shift per user, per session, per task. A consumer might use the same AI assistant for budgeting today, mortgage comparison tomorrow, and tax optimization next quarter. Each task requires different data.

Scope fluidity is not entirely new. Multi-purpose applications today already manage varying scope needs – a data recipient with multiple use cases may request all possible scopes even if a user only exercises a subset in any given session. What is qualitatively different about agents is that the purpose is not just broad but emergent: it is defined by the consumer's questions in real time, is not fully known at the time of app registration, and may evolve in ways that neither the developer nor the consumer anticipated when consent was originally granted.

> ***Illustrative example:*** *An AI financial assistant is initially authorized to access a consumer's checking account transactions for budgeting. The consumer then asks the assistant to "help me figure out if I'm paying too much for insurance." The*

> *assistant determines it needs the consumer's full account history, including policy payment transactions it hasn't been scoped to access. Under a traditional fixed-purpose app, this would require a new registration and consent flow. But the assistant is one application serving many purposes as its scope needs are as fluid as the consumer's questions.*

This fluidity makes it significantly harder to match data scopes to a defined purpose and to apply data minimization principles, as it multiplies across registration, consumer consent, and consent modification flows. FDX has historically defined scopes at the data-cluster level. The open question is what, if anything, about agents requires evolving beyond that model.

A risk: If the API-based permissioned-access path cannot accommodate the fluidity that agents require, agents may route around it by reverting to credential-based scraping (Use Case 3) to access data that the API's scope model does not easily provide. This could reintroduce many of the risks that FDX standards were created to address.

3.4 Provenance, Auditability, and the Third-Hop Problem

Applies to: Use Cases 1 and 2 (API-mediated access).

In some agentic architectures, an agent connects to a tool (such as an MCP Server) to access FDX APIs, then sends retrieved financial data to a third-party LLM for processing. Open banking implementers have often wrestled with downstream data visibility, and agentic AI may amplify this challenge: in some architectures, data flow to a third-party processing service is central to how the system operates.

In some cases, agents use structured tool calls and sanitized queries rather than sending raw financial data to third-party LLMs. Many agents can perform categorization, analysis, and reasoning by passing structured requests to the model – receiving back instructions on what action to take – without exposing the underlying consumer data. Additionally, on-device and self-hosted models are becoming increasingly viable, enabling agent reasoning to occur locally without data leaving the consumer's environment or the data recipient's infrastructure.

When consumer financial data ***does*** travel through a party that is not contracted or registered under the existing data-sharing framework, the governance chain breaks. The processing service becomes a participant in the data flow that existing standards, regulations, or business terms may not reach. How can governance frameworks encourage and reward better architectural patterns – while maintaining accountability for cases where data does leave the controlled chain?

A related consideration:

Deterministic vs. probabilistic processing

FDX's standards have historically assumed that data processing is deterministic: given the same input, a system produces the same output. LLMs introduce a probabilistic element – the same financial data may be interpreted, summarized, or acted upon differently each time it is processed. However, it is important to note that the actions that matter for FDX's standards (e.g., API calls, scope requests, data retrieval) remain deterministic. The LLM serves as a reasoning layer that decides which structured action to take, but the data standard itself does not need to change because the decision-maker is probabilistic. The open question is narrower: can scope frameworks and audit mechanisms designed for fixed-purpose applications accommodate the fluid, LLM-driven reasoning that determines what data to request and how to use it?

> ***Illustrative example:*** *A fintech's AI agent retrieves a consumer's transaction data through an FDX API. In a less mature architecture, the agent sends raw data to a cloud-hosted LLM for categorization. The LLM provider is not a registered data recipient under the fintech's agreement with the bank. In a better-designed architecture, the agent sends structured queries to the LLM ("categorize this merchant code") without exposing the consumer's actual transaction data. The design question for standards bodies and ecosystem participants is: how do we encourage the latter pattern through technical standards and establish accountability for the former?*

3.5 Revocation and Data Retention

Applies to: All three use cases.

Revoking an API token terminates future data access – it stops new data from flowing. This is operationally distinct from data deletion, and the distinction is not new to agents. When a consumer revokes a traditional app's access today, the app retains data it has already received – a consumer who disconnects their bank from a payments app likely does not expect it to forget their account and routing number. The consumer may continue to use the app with the data it already has.

Agents do not change this fundamental principle. However, agents may introduce new ambiguity about what "data already received" means. If an agent has incorporated financial data into persistent memory (such as conversation histories, embedding stores, or derived summaries), or used it to generate derived insights, the boundary between "retained data" and "processed knowledge" becomes less clear than it is for a traditional database record.

> ***Illustrative example:*** *A consumer revokes their fintech app's access to their bank account. The API token is terminated and no new data can be retrieved. But the fintech's AI agent has already incorporated six months of transaction data into a persistent "memory" – a vector store that the agent references when answering consumer questions. The derived summaries, spending patterns, and financial insights remain in the agent's context. Locating and deleting this information is architecturally different from deleting a database record.*

[Note: Historically, FDX technical standards have not focused on data retention or deletion requirements. Whether agentic AI creates a need for new industry standards or guidance, or whether these topics remain the province of contracts and regulation, is an open question.]

3.6 Security: Prompt Injection, Tool Misuse, and Abuse Amplification

Applies to: All three use cases, with distinct threat profiles for each.

Existing fraud controls assume human or application-driven behavior with known patterns and thresholds. Agentic AI introduces attack vectors that are qualitatively different from traditional API security threats.

Prompt injection: Malicious inputs can manipulate an agent into acting outside its authorized boundaries, revealing data it should not, or misrepresenting its actions to the user.

Abuse amplification: Agent behavior may appear individually legitimate but, in aggregate, exceed the intent of the original authorization. Autonomous agents can scale actions (e.g., data collection, account probing, transaction initiation) far beyond what a human user could do manually.

Adaptive controls gap: Current enforcement is typically static or reactive. Adapting controls dynamically based on delegation context, agent behavior patterns, and real-time risk signals is an unsolved problem.

> ***Illustrative example (adaptive controls gap):*** *A legitimate AI agent authorized to access a consumer's checking account begins querying the API every five minutes instead of once daily, pulling full transaction histories on each call rather than incremental updates. Each individual request is within the authorized scope. But the aggregate behavior – high-frequency, full-history pulls – resembles a data-harvesting pattern that a static rate limiter may not catch because each call is technically compliant. A data provider needs the ability to detect and respond to behavioral anomalies in real time, adjusting controls based on delegation context rather than just request-level validation.*

After reviewing this Brief, we invite industry stakeholders to:

1. **Share your input** on the accompanying **Call for Input** question by Friday, May 29, 2026
2. **Get involved with FDX's discussions** on Agentic AI & Open Finance with other industry leaders. [Contact FDX for more information.]

Appendix

Appendix 1: External Standards Landscape

The agentic AI standards landscape is evolving rapidly. The following is a point-in-time snapshot, not a comprehensive survey, intended to help readers understand where relevant work is underway and where FDX's contributions would be additive rather than duplicative. We continue to monitor this landscape and welcome input on developments not captured here.

A. Organizations and Initiatives

The following organizations are engaged in work that intersects with or is adjacent to FDX's agentic AI inquiry. For each, we note their focus and current status, as well as where FDX's work would fill a gap or complement their efforts. This is a point-in-time snapshot as of March 2026.

Organization	Focus & Status	Gap / Opportunity for FDX
NIST NCCoE	Enterprise agent identity and authorization demonstrations. Feb 2026 concept paper on agent identity standards. Enterprise-focused.	Enterprise-focused; does not address consumer-permissioned financial data flows or the consent/delegation challenges specific to open banking.
OpenID Foundation / FAPI	Financial-grade API security profiles; FAPI 2.0 baseline. AI Identity Management Community Group (AIIMCG) has published on Identity Management for Agentic AI. Proposed OIDC-A extension.	Relevant to FDX's security model. Not yet addressing agent-specific delegation patterns. Potential alignment point for FDX-specific agent security profiles.
Agentic AI Foundation (Linux Foundation)	Neutral governance for MCP, Goose, AGENTS.md. Agent-to-tool connectivity and interoperability standards.	Cross-industry focus; no financial-services-specific risk, consent, or data governance layers. FDX defines the governance layer that agents access through these protocols.
Agent Commerce Protocols	Google AP2/UCP, Visa Trusted Agent Protocol, Mastercard Agent Pay, Mastercard/Google Verifiable Intent, Stripe/OpenAI Agentic Commerce Protocol. Focused on agent-mediated transactions.	Payment-focused rather than data-access-focused. Adjacent to FDX's scope; relevant as agents cross from read to write operations.
W3C	Verifiable Credentials standard (relevant to agent attestation). AI Agent Protocol Community Group formed.	General-purpose identity and credential frameworks. FDX could profile VCs for financial-services-specific agent attestation.
ISO/IEC 42001	AI management systems standard. Organizational governance requirements for AI deployment.	Organization-level, not protocol-level. Complementary to FDX's technical standards work.
FSSCC / Cyber Risk Institute	FS AI Risk Management Framework (FS AI RMF). Financial-services-specific AI risk management guidance.	Risk-management-oriented rather than interoperability-oriented. Relevant to how FDX members manage agent deployment risk internally.

Future of Privacy Forum	Center for AI: researching data use and privacy implications of AI systems.	Privacy and data-use focused. Complementary to FDX's work on consent and data minimization in agentic contexts.
UK Open Banking Limited	AI and Open Finance green paper exploring similar questions. Regulatory-driven with FCA oversight.	UK/EU regulatory context. Parallel exploration; useful comparative reference but different regulatory regime.
Agentic Futures Initiative	Cross-sector policy coalition (Anthropic, Intuit, others). Focused on policymaker education and responsible agent deployment.	Policy-oriented, not standards-oriented. Potential engagement channel for FDX.

B. Protocols in the Landscape

The following protocols are frequently cited in the agentic AI conversation. For each, we note its function and how it relates to FDX's scope.

Agent-to-system interaction:

Model Context Protocol (MCP): Developed by Anthropic, now under Linux Foundation governance. An application-layer protocol that defines tool discovery, capability negotiation, and authorization patterns for AI agents interacting with external data sources and tools. OAuth 2.1 integrated for authorization. *FDX relationship: FDX defines the financial data standards and consent requirements that agents must follow when accessing data through MCP-connected tools and similar integration patterns.*

WebBotAuth: Developed by Cloudflare. Foundational authentication layer for distinguishing between types of automated web clients. *FDX relationship: Relevant to Use Case 3 (credential-based access) and the agent identification challenge.*

Agent commerce and payments:

A2A (Agent-to-Agent Protocol): Led by Google. Enables agents to communicate and transact with each other. *FDX relationship: Adjacent; focused on agent-to-agent interaction rather than agent-to-data-provider access.*

Unified Commerce Protocol (UCP): Co-developed by Google, Shopify, and others. Defines standardized interactions for commerce across agent platforms. *FDX relationship: Commerce-focused. Relevant as agents cross from data access to financial actions.*

Agentic Commerce Protocol (ACP): OpenAI and Stripe. Enables agents to make purchases on behalf of users. *FDX relationship: Payment-focused. Relevant as agents cross from data access to financial actions.*

Trusted Agent Protocol (TAP): Visa and Cloudflare. Provides identity verification for agents engaged in commerce. *FDX relationship: Agent identity verification for transactions; potential alignment point for FDX agent attestation work.*

KYAPay: Skyfire. "Know Your Agent" payment infrastructure for autonomous agents. *FDX relationship: The KYA concept is relevant to FDX's agent identification problem space, though KYAPay itself is payment-focused.*

Verifiable Intent: Mastercard and Google. An open standard enabling agents to cryptographically express and verify a consumer's intent for a transaction, allowing merchants and payment providers to validate that an agent is acting on an authenticated instruction. *FDX relationship: Directly relevant to the consent provenance challenge – how downstream parties verify that an agent's action traces back to a genuine consumer decision. Potential alignment point for FDX work on agent delegation.*

Agent Pay: Mastercard. A payment credential framework designed for agent-initiated transactions. *FDX relationship: Payment-focused; relevant as agents cross from data access to financial actions.*

Key finding: No external body currently defines technical standards specifically for AI agents accessing consumer-permissioned financial data through standardized APIs. Generic agent infrastructure, identity, and commerce protocols are maturing rapidly. But the intersection of AI agents with permissioned financial data access – consent models, data minimization requirements, agent traffic classification, and downstream data handling specific to financial services – remains an open governance space.

About FDX's Agentic AI Task Force

In late 2025, FDX's governing body chartered a time-bound Agentic AI Task Force to investigate these questions. The task is open to all FDX members, with active recruitment across financial institutions, data access platforms, fintechs, risk and compliance leaders, AI practitioners, and technology providers.

FDX is actively working to bring non-member organizations into this conversation – particularly AI platform operators, identity standards bodies, and policy organizations whose perspectives are essential.

[i] Endnote

Out of scope topics for this Brief include:

- Defining guidelines for how AI large language models (LLMs) are designed or trained
- Defining broad AI ethics frameworks
- Providing guidance on who is or should be legally liable when an AI agent makes an error, causes financial harm, or acts outside its authorization
- Defining baseline quality or trustworthiness criteria for AI agent[i]s.

Adjacent questions: AI is impacting the Open Finance landscape in many ways beyond the scope of this brief; for example, more companies are using AI internally to build APIs, detect fraud, process user-permissioned financial data, generate insights, and create new customer experiences. These uses of AI raise important questions about data deletion, data use, liability, third-party onboarding, and risk management that warrant attention through industry dialogue, bilateral relationships, regulation, or other frameworks. Because these activities do not directly involve ***how*** permissioned data flows between parties, they are not a primary focus of this inquiry.

The regulatory and legal landscape: This brief acknowledges that the regulatory environment has not kept pace with the rapid evolution of agentic AI. There are material unresolved questions from a legal perspective that will inform the standards work described here. These include:

- the legal status of an AI agent (is it a "user," a "service provider," or a "separate actor");
- whether and how legal consent frameworks designed for human-initiated, single-purpose interactions apply when an agent acts autonomously across fluid purposes; and
- how existing privacy, data protection, and financial services regulations in different jurisdictions (including U.S. federal and state law, EU/GDPR, and emerging Canadian open banking requirements) will adapt to agentic patterns.

FDX will monitor these legal issues as context for standards work.