Sahamati Labs
June 2026

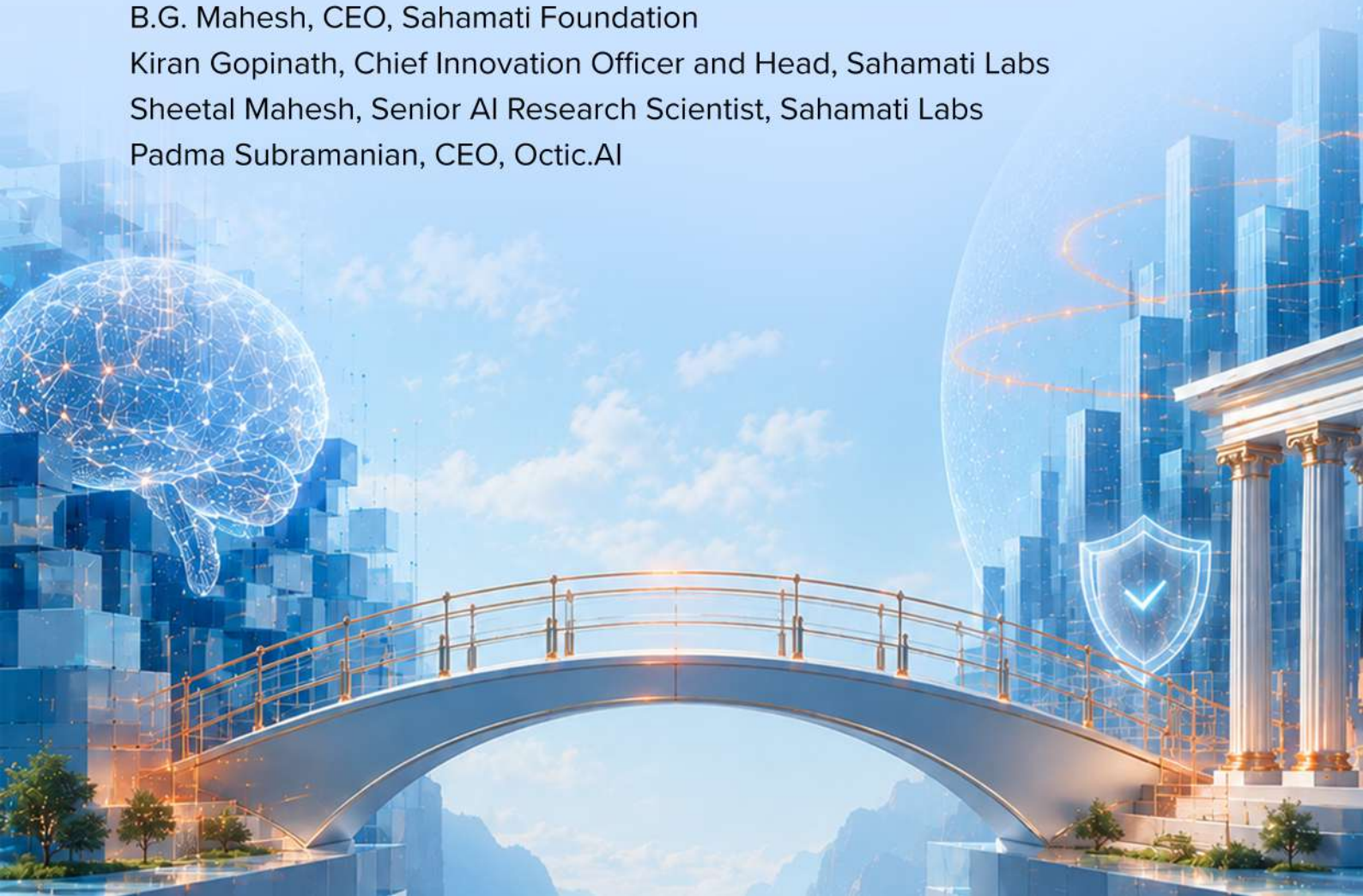
Sahamati®

A Framework for AI Agents in the Account Aggregator Ecosystem

Four Pillars of Consent, Identity, Provenance, and Conformance Built on the Existing AA Foundation

Authors:

B.G. Mahesh, CEO, Sahamati Foundation
Kiran Gopinath, Chief Innovation Officer and Head, Sahamati Labs
Sheetal Mahesh, Senior AI Research Scientist, Sahamati Labs
Padma Subramanian, CEO, Octic.AI



Acknowledgements

Our sincere thanks go to the following people whose time, dedication, insights, expertise, and collaborative spirit were instrumental in shaping the ideas presented here.

Contributors:

Geethashree Srikanta, Senior Lead Governance, Sahamati Foundation
Pranav Narain, Head Legal and Compliance, Sahamati Foundation
Rene Kolga, Product Leader, Google
Shalini Gupta, Chief Policy and Advocacy Officer, Sahamati Foundation
Suraj Moraje, Independent Board Director, Sahamati Foundation

With:

Adrita Chakravorty, Research Analyst, Sahamati Labs
Darshini Konsoor, Corporate Communications Manager, Sahamati Foundation
Piyali Chatterjee, Marketing and Communications Specialist, Sahamati Foundation

About Sahamati Labs

Sahamati Labs is the research and innovation initiative of Sahamati, the Self Regulatory Organisation (SRO) for India's Account Aggregator ecosystem. It advances research, develops reference architectures and explores emerging technologies that will shape the future of Open Finance and trusted digital financial infrastructure.

Working at the intersection of artificial intelligence, Open Finance, privacy enhancing technologies and consent based data sharing, Sahamati Labs collaborates with banks, NBFCs, fintechs, technology providers, researchers, policymakers and other ecosystem participants to shape the future of trusted AI and Open Finance. Through applied research, ecosystem collaboration and open design, the Labs develops frameworks that enable innovation while strengthening trust, interoperability, accountability and consumer protection.

Table of Contents

Executive Summary

The Account Aggregator (AA) framework rests on a promise to the Customer: their financial data flows only under their consent to regulated entities the regulator has accredited, and what those entities do with the data is governed by law. AI agents are poised to enter the ecosystem as the new layer that reads and acts on consented data on behalf of the Financial Information Users (FIUs). Banks, lenders, and advisers are preparing to use AI-driven software to classify income from bank statements, screen for fraud, and recommend or make credit decisions. The AA specification does not cover this processing layer.

This paper proposes four extensions to the AA primitives the specification already defines, organised as four pillars:

- **Consent:** Makes the intended use of the Customer's data explicit, including how the data may be processed and what outputs may be produced.
- **Identity:** Every agent operating on AA data is registered and known to the ecosystem.
- **Provenance:** Signed records of every agent action, visible to the Customer in plain language and generated by infrastructure the agent cannot suppress.
- **Conformance:** Pre-production certification against known attack patterns.

The framework rests on a shared infrastructure model with two roles: a Governance Operator and a Runtime Operator.

An industry body operates the governance layer (the registry, the audit trail, the safety-testing library) and sets the rules these encode.

A separately designated entity, under regulatory oversight, operates the protected runtime that holds and processes AA data in a Trusted Execution Environment (TEE), sometimes called a Confidential Clean Room (CCR) in India. The Financial Information Provider (FIP) encrypts and dispatches the data to the TEE (protected runtime), where it is decrypted and processed exclusively within the secure enclave. Following processing, the protected runtime re-encrypts the dataset to the FIU's key for the FIU to retain as per applicable regulatory and legal requirements.

Agentic processing runs only inside the protected runtime, not on the FIU's copy. Every receipt the framework relies on is generated by that runtime, not by the agent or the FIU, so what gets recorded is independent of the party whose actions are being recorded.

The framework is the smallest set of extensions that re-establishes ecosystem-wide accountability for AI processing without restricting innovation. It depends on ecosystem direction on four points: acceptance of the shared infrastructure model, designation of the TEE operator (which the framework leaves as a regulatory choice), recognition of the registered-agent model as a compliance basis, and treatment of the audit trail as a regulatory artefact.

Shared trust infrastructure for agentic processing reduces the cost of trustworthy deployment for the ecosystem as a whole and enables broader AI adoption by establishing the confidence required for institutions, regulators, and consumers to participate at scale.

Sahamati Labs is publishing this paper for ecosystem engagement and consideration.

Feedback on this paper may be sent to labs@sahamati.org.in

1. The Problem Statement

The Customer is the main entity around which the entire AA framework is built upon. Under the current model, Customers provide consent for their financial data to be used solely by regulated entities, who are accredited to ensure compliance with applicable regulations and legal requirements. This ensures that there is a purpose limitation and Customers are protected against financial fraud, phishing or identity theft. However, AI agents are entering the ecosystem as the new layer that reads and acts on consented data inside FIUs. Essentially, this empowers banks, lending institutions, and financial advisers to use AI-driven software to classify income from bank statements, screen for fraud, and recommend or make credit decisions.

FIUs are preparing AI driven underwriting, fraud detection, and collections agents. While this enables FIUs to make more informed decisions, it may undermine customer consent if the newly embedded AI agents process data in ways that extend beyond the purpose scope for which consent was obtained under the AA framework.

This not only weakens customer trust in the AA framework but also obscures the processing activities performed by the FIU's AI agents, reducing transparency into how customer data is used, constraining independent auditability, and weakening the mechanisms required to establish accountability and redress.

Moreover, the AI-enabled processing layer heightens the "black-box" issue, wherein there is no clarity on how the AI system uses the data and arrives at certain conclusions. This makes auditing a difficult task and results in trust deficit for users. This also leads to loss of control as the industry body cannot easily verify the process, leaving significant loopholes in the secured financial environment. Interestingly, even the FIUs deploying the AI systems may not be aware of the scope and particular steps taken by AI agents, resulting in process ambiguity. Hence, a transition into a new framework using a TEE is the need of the hour. By employing this new framework, the narrative can be shifted to user safety, privacy, and protection against misuse.

2. What Does the Framework Do?

The new framework rethinks existing models and tries to accommodate new AI developments within its ambit by using a TEE. This is done by extending the existing AA primitives of Consent, Identity, Provenance and Conformance, which, if done right, can re-establish ecosystem-wide accountability for AI processing without restricting innovation.

In this, emphasis is laid upon four main points of regulatory oversight, namely, acceptance of the shared infrastructure model, designation of the TEE operator (which the existing framework leaves as a regulatory choice), recognition of the registered-agent model as a compliance basis, and treatment of the audit trail as a regulatory artefact.

Here, Confidential Computing via a TEE emerges as a means to ensure the data processing of AA data can be tracked through processing receipts, conformity to attested and consented use, refusal for unconsented use, and prevention of unattested agents from processing data. The framework also accounts for user revocation and stops processing of AA data in runtime. Overall, this framework seeks to revamp the existing model to be better equipped to serve the new challenges. In this section, the existing framework along with the proposed tweaks in the AA ecosystem is detailed.

2.1 An AA Native Approach

The current AA ecosystem establishes a process whose integrity is verifiable end to end. A signed consent artefact, signed data receipts, and a participant registry of accredited entities together produce an ecosystem grade trust without requiring any party to inspect any other party's internal systems.

Applied to agentic processing, the same approach suggests something distinct from runtime instrumentation, and complementary to it. The framework does not attempt to second guess the agent's reasoning in real time. The framework does not attempt to restrict what an agent can do. Instead, it ensures that the agent operates within a trusted and accountable environment

The agent must act under a consent that explicitly authorises the type of processing being performed. It must run as a registered agent class within an attested TEE, with its capabilities and behaviour declared in advance.

Every action taken by the agent is independently observed and recorded by the TEE through signed processing receipts. In addition, the agent class must be certified against known attack patterns and failure modes before it is allowed to operate within the framework.

The agent may still make incorrect decisions or arrive at flawed conclusions. The difference is that those decisions are now visible, traceable, and auditable, rather than remaining hidden within an FIU's internal systems.

This catches purpose drift, lineage breaks, scope violations, and unregistered processing structurally rather than statistically. It does not catch a real time prompt injection at the exact moment of attack, but it does catch the certification gap that allowed the injection to succeed, and forces re-certification when new injection classes emerge. However, the runtime tooling is what individual FIUs and TSPs have to build inside their own perimeters. This specification extension framework also mandates an industry body to lead in the ecosystem.

At this stage, it becomes critical to trace AA data usage, especially with reference to what happens to fetched financial data once it reaches the FIU, who processes it, under which authority, and with what derivatives. In this, the AA's data-blind role, wherein the AA cannot see or process the FI data is not altered but reshaped by giving it a new consent-management and routing work. This means that the data remains untouched and Processing Aware Consent (PAC) blocks are now embedded in the consent artefact issued by the AA, thereby allowing the consent-capture to record the processing authorisation and not just the data scope. Additionally, the AA routes the signed consent artefact to the TEE alongside the data and passes the data reference to the FIU while rendering a customer receipt to the Customer in plain language. Here, it is worth noting that these are additions to the AA's consent-manager surface, not departures from its data-blind position.

Critically, this framework largely focuses on the FIU side, with little to no change for the FIPs. The FIP's role here is operational, not agentic, therefore, it encrypts AA-fetched data under the AA-issued consent and dispatches it as today, with one change: the recipient in TEE is operated under regulatory oversight rather than the FIU. The TEE decrypts inside the enclave, hosts agentic processing on its copy, and re-encrypts the dataset to the FIU's key for the FIU to retain as per applicable regulations and laws.

2.2 The FIP's Role

Presently, the FI data is encrypted by the FIP, which then moves to the AA, before eventually being decrypted by the FIU. So the data crosses multiple network hops and briefly becomes cleartext at the FIU's perimeter. In this framework, the FIP's wire behaviour is exactly the same but what shifts is the recipient behaviour.

Similar to the current model, the FIP encrypts and dispatches a single ciphertext, but now, it is encrypted to the public half of a keypair generated inside the ecosystem's TEE, with the private half held in the hardware-protected enclave memory that nobody outside the enclave can access: not the TEE operator, not the FIU, not the FIP.

The TEE is the only entity that can decrypt the ciphertext. The AA remains in the path as the

router, as it is today, but it routes a ciphertext encrypted to the TEE that it cannot read, so its data-blind role is preserved on the routing as well.

a) Two Copies with Varying Lifetimes

Inside the enclave, the TEE holds the dataset for agentic processing under the consent, and re-encrypts a copy to the FIU's public key for the FIU to retain as per applicable regulatory and legal requirements. The FIU's public key is the one registered against its accredited profile in the Central Registry; the TEE fetches it at processing time, so no new key-distribution channel is introduced. The re-encryption happens inside the enclave; cleartext bytes never leave it. The FIU receives an encrypted blob it can decrypt with its own key, the same way it receives encrypted data today, except that the source is the TEE rather than the FIP and the encryption is performed by the TEE.

AI agent processing on AA data does not run on this copy; it runs only on the copy inside the TEE. The FIU's copy is the retention copy. The FIU stores it under the controls and retention period the regulator requires.

The data reference and the retention copy have different lifetimes, and the distinction matters. The reference is access to the cleartext the TEE holds: when the consent expires or is revoked, the TEE stops honouring the reference and discards its cleartext copy, so agentic access ends at consent end. The FIU's retention copy persists beyond that, but only for as long as the applicable regulations and laws require, and only for the non-agentic purposes the consent authorised; it is governed by the FIU's deletion and retention rules, with deletion receipts emitted as proof when it is disposed. Ending the consent terminates agentic access immediately; it does not by itself erase a retention copy the law requires the FIU to keep.

b) The Need to Route the Data Through the TEE

The primary reason is that the FIU's perimeter also applies to the FIU, and AI processing happening there is invisible to anyone outside. So routing the dispatch through the TEE places controls at runtime in the agentic processing inside attested infrastructure beyond the purview of any agent, including the operator. The cleartext exists only inside that runtime and only for the lifetime of the consent. The enclave's keypair is bound to the runtime measurement, so any tampering with the runtime changes the measurement and the keypair. It also makes any ciphertext encrypted to the previous public key undecryptable. Essentially, tampering destroys access rather than enabling it.

c) Key Rotation

A change in a TEE's software, whether due to a security patch, upgrade, or failover to a backup environment, results in a new attestation measurement and a new cryptographic keypair. The framework treats these as expected operational events and manages them through an explicit key rotation process.

When a new enclave version is deployed, its attested public key is published and made active before the previous key is retired. New data is automatically encrypted to the active key, while data already associated with the retiring key continues to be processed by the older enclave version or is securely re-encrypted within the enclave and transferred to the new version. This ensures that no active consent, processing task, or audit trail is disrupted during the transition.

The framework also maintains a verifiable record of the attested key active at any point in time. As a result, historical actions remain auditable against the specific enclave instance and key that were valid when those actions occurred.

Here, the industry body plays a significant role by retaining the history of valid enclave measurements and their public keys in the governance layer with a specified validity window. This allows the regulator verifying the attestation that was current at the time of a past action to check it against the measurement that was live then, and not only the measurement that was live on the day of verification.

d) The FIP's Responsibility

The FIP's responsibility is to verify that the public key it is encrypting belongs to a genuine, untampered enclave before encryption. If that verification fails, because the attestation is stale, the measurement does not match a published value, or no active enclave key is available, the FIP fails closed. So it does not encrypt to an unverified key and does not fall back to dispatching cleartext or to an alternate recipient. In fact, a failed verification stops the fetch, the same way an invalid consent stops a fetch today, rather than switching to a less safe path.

Since the FIP's wire behaviour is unchanged (encrypt and dispatch one ciphertext as today), the FIP itself takes on no new availability dependency. The TEE is on the path between the FIP and the FIU's retention copy. If the TEE is unavailable when the dispatch happens, the ciphertext queues at the TEE's ingress, agentic processing waits, and the FIU's retention copy is forwarded once the TEE recovers. Therefore, the framework creates dependency on the TEE being available to process and forward, which is mitigated by multi-region TEE deployments with attested failover (Appendix A).

2.3 The FIU's Role

The FIU remains accountable as per applicable regulations and laws to Customers for decisions made on their data. What the framework changes is where the agentic processing that produces those decisions happens, and what evidence the FIU registers, retains, and stands behind. The FIU continues to hold the data it is legally responsible for holding; the agents that read and act on the data run inside infrastructure operated either by the framework's designated TEE operator or, where the FIU has chosen to do so, by the FIU itself in its operator capacity (Section 4). The FIU's accountability is met through registration, certification, and the receipt chain rather than through internal-log assertion.

a) Agent Class Registration

Before any agent runs on AA data on the FIU's behalf, the FIU registers each agent class in the Agent Class Registry. The manifest declares identity, served principals, capabilities, tool surface, model bindings, permitted children, side-effect classification, and the class's current Conformance certification status. The FIU may register classes it has built itself or classes built by a TSP. In the TSP case, the TSP submits the manifest with the FIU named in served principals, and the FIU counter-signs to activate the binding. Registration is at class granularity: an FIU may run many instances of a registered class, but the class is the unit of identity, certification, and accountability.

b) Processing Agreement with the TEE Operator

The FIU's relationship with the designated TEE operator is the processing agreement that establishes the operator's role as Data Processor as per applicable regulations and laws. The agreement is a registration prerequisite, alongside the operator's accreditation. The framework does not displace the FIU's accountability as per applicable regulatory and legal requirements; the operator processes on the FIU's behalf, under the consent the FIU's flow obtained, and under the constraints attestation enforces.

c) Retention Copy

The FIU receives a retention copy of the consented data, re-encrypted by the TEE to the FIU's public key and forwarded under attestation. The retention copy is the dataset agentic processing actually acted on, with the TEE's signature on the re-encryption as evidence of provenance. The FIU stores it under the controls and retention period the regulator requires. Agentic processing does not run on this copy; it runs only on the TEE's copy under the active consent. The retention copy may persist beyond consent end for the obligations applicable regulations and laws impose, but only for non-agentic purposes the consent authorised, with deletion receipts generated as proof when it is disposed.

d) Decision Dossiers

For each decision an agent makes on the FIU's behalf, the TEE returns the decision output and its supporting record to the FIU at the end of processing, alongside the retention copy. The FIU holds these decision dossiers inside its own perimeter as the accountable party for the decision. The system receipts the TEE emits to the Processing Receipt Registry reference and address these dossiers without exposing their contents, which means the FIU's record of how a decision was made is externally addressable and verifiable without the underlying detail leaving the FIU.

e) Grievance Handling

The FIU operates the grievance surface for disputes about decisions made on its behalf. The Customer presents the consent identifier and the decision in dispute. The FIU's grievance officer follows the receipt chain from the customer receipt through the system receipts to the dossier the FIU holds, and inspects the prompts, tool calls, and policy checks that produced the disputed decision. The framework's contribution is that the evidence the grievance officer works with is structured and externally verifiable rather than reconstructed from internal logs.

f) Revocation Handling

When a Customer revokes a consent at the AA, the revocation propagates to the TEE: agentic processing on the FIU's behalf under that consent ends, in-flight invocations complete with termination receipts, and new attempts produce refusal receipts.

The FIU's retention copy persists under applicable regulatory and legal retention obligations but cannot be re-processed agentically. Where an agent has decided but the FIU has not yet executed the decision through its downstream systems, the FIU's existing processes govern whether to proceed or hold; the framework does not reach the FIU's own systems.

2.4 The AA's Role

The AA's wire role is largely unchanged: it remains the RBI-licensed consent manager intermediating between FIPs and FIUs, and its data-blind position is preserved. What the framework adds is on the consent-management surface, not on the data-handling surface.

The signed consent artefact the AA issues now carries a PAC block alongside its existing scope and purpose declarations, listing the permitted agent classes, the processing classes from a controlled vocabulary the industry body publishes, the derivatives that may be retained and for how long, and the transformations explicitly out of scope. The AA's consent-grant flow surfaces these to the Customer so the grant or refusal is made on processing terms, not just data terms.

On dispatch, the AA routes the ciphertext to the TEE rather than to the FIU, alongside the signed consent artefact the TEE will enforce against. The ciphertext is encrypted to the TEE's enclave-bound public key, which the AA cannot read, so the data-blind position holds on routing as it does today. The data reference the TEE issues for the retention copy is passed by the AA to the FIU.

The Customer views the customer receipt for each consent through the AA's consent dashboard, fetched from the Processing Receipt Registry by consent handle. The Processing Receipt Registry releases customer receipts only to the AA that issued the consent, so consumer-identity resolution stays at the AA: no other party in the framework can resolve a handle to an identified Customer. Revocation by the Customer at the AA propagates to the TEE, which stops honouring the data reference and emits refusal receipts for any new attempts.

3. The Framework

In order to protect the Customer, there is a need to employ both old and new means. On one hand, it is important to identify what already works. On the other hand, it is important to discover new means to penetrate the processing layers and make them accountable. Therefore, in this context, going back to the foundational pillars is a good starting point.

3.1 The Four Pillars

Four pillars hold up the framework. Each takes one part of what safe agentic processing should mean and implements it as an extension to an AA primitive that already exists.

Table 1: The Four Pillars: Extensions to Existing AA Primitives

Pillar	Existing AA Primitive	Extension	What the Pillar Establishes
Consent	Consent artefact, with its purpose code and data life fields	Processing Aware Consents	What the FIU is permitted to do with the data, not just what data may be fetched.
Identity	Industry-body Central Registry of accredited FIPs, FIUs, and Consent Managers	Agent Class Registry	Who is permitted to act. Every agent class is uniquely identifiable, registered against the operating FIU, and certified.

Pillar	Existing AA Primitive	Extension	What the Pillar Establishes
Provenance	Consent receipts and data receipts (the network logs that act as receipts today)	Processing Receipts	What actually happened. Every agentic action emits a signed, auditable receipt referencing the consent, the agent class, the source artefacts, and the tool calls.
Conformance	Industry-body certification program for AA participants	Conformance	Whether the agent stays within its declared manifest under stress. Standing requires passing the current Library and its versions, which the industry body maintains.

Consent expresses what is allowed. Identity says who is allowed to do it. Provenance records what actually happened. Conformance verifies that the system as a whole behaves as declared.

Together they produce an ecosystem grade evidence that an agent operating on AA data is governable, audit ready, and structurally bounded, without the industry body operating anything inside an FIU's perimeter.

The runtime that holds processing is the TEE, which sits outside every participant's perimeter and is operated by a separately designated entity, while the industry body operates only the governance layer.

Two clarifications about scope before the mechanics. First, AA itself is read only by design: FIUs fetch FI data through consent mediated flows, and there is no write path from an FIU back to an FIP through AA. Since every FIU only ever reads from the source, the agent tiers below are not about whether an agent reads (they all do), but they are about what an agent does with what it reads. Second, this framework governs what happens within AA's scope only.

The agent may also call non-AA systems (its own internal services, third-party services) under whatever regimes those systems impose, but those interactions sit outside the framework and the receipt model. The classification axis below describes agent class capabilities exercised on or after AA mediated reads: derive only (the agent produces derivatives such as summaries or scores from the data but changes no state) or internal action (the agent changes state in the FIU's own systems such as its CRM or loan origination system).

3.1.1 Consent

From the Customer's perspective, consent in the proposed framework still does what it did before: it authorises a specific FIU to receive specific data for a specific purpose, for a specific duration. What changes is that the consent now also authorises (or refuses) AI processing of that data. The artefact sent by the AA carries a PAC block that declares what the FIU is permitted to do with the data, alongside the existing scope and purpose declarations. The PAC block lists the permitted agent classes by their Agent Class Registry (ACR) identifiers, the processing classes from a controlled vocabulary that the industry body publishes, the derivatives that may be retained and for how long, and the transformations that are explicitly out of scope. The data life declared in the consent remains the outer bound; the PAC block cannot extend it.

Every consent that authorises AI processing of AA data, derive only or consequential, requires a WebAuthn assertion at grant time, bound to the consent payload, including the PAC block, in addition to the standard AA flow. The framework does not treat derive only processing as low stakes enough to authorise on a weaker factor: reading and deriving from a person's financial data is itself processing the Customer should strongly and verifiably authorise.

The practical constraint is authenticator availability, since not every Customer has a WebAuthn capable device today; the framework's requirement is uniform WebAuthn, and a regulator may choose to phase it as authenticator coverage grows, but the target state is that no AI processing of AA data proceeds on anything weaker than a WebAuthn bound consent.

a) The Consent Lifecycle

Revocation propagates through all four pillars. The consent status flips from active to revoked. Since all agentic processing runs inside the TEE, the TEE checks consent status before every action by every agent and refuses to start one under a revoked or expired consent, emitting a refusal receipt. Revocation does not depend on any agent's own good behaviour, and there is no class of agent for which it degrades to a status check because the agent is merely trusted to perform. So essentially, every event in the lifecycle of a consent emits a signed Processing Receipt.

b) Strong User Binding: WebAuthn for Every Consent

AA authentication today rests on OTP. A Customer proves possession of a registered mobile number to authenticate against their AA at consent time. This was a reasonable choice for a network whose principal risks were intermediary error and disputed consent. It is a less

reasonable choice for a network whose principal risks are now, agents acting on FI data, decisions being attributed to Customers who never authorised the specific processing, and grievance review against agentic decisions.

WebAuthn provides phishing resistant user binding, and the framework requires it for every consent that authorises agentic processing of AA data, not only consequential ones. The Customer's authenticator (passkey or hardware key) produces a cryptographic assertion bound to the specific consent payload, including the PAC block. The assertion travels with the consent handle, providing structured evidence that the Customer actually saw and authorised the processing this consent authorises, including which agent classes may operate under it.

Consent fatigue is the failure mode this framework most needs to resist. The principle is to prompt on transitions and policy boundaries, not on every action. Essentially, consent authorises a class of processing for a bounded period and the agent acts within that class without re-prompting. Human in the loop step up is reserved for irreversible or precedent setting cases.

3.1.2 Identity

Identity is an indispensable part of authorisation as it decides who can or cannot access the data. In this, the ACR is the binding fabric. Processing Aware Consents reference agent class identifiers that resolve to the ACR. Processing Receipts anchor identity to ACR entries. Conformance certifies entities recorded in the ACR. The ACR is the first piece of operational infrastructure the framework requires the industry body to stand up.

a) Agent Class Registry Content

Each agent class is registered as a signed manifest declaring: identity (agent class name, deployment fingerprint), the served principals (the FIU identifiers the agent class is authorised to act on behalf of, each of which must counter sign before the binding is active), capabilities (data scopes the class is permitted to read, processing classes from the controlled vocabulary, side effect surface), tool surface (the registered MCP servers it may invoke), model bindings (which models and at what version), permitted children (the list of child agent class identifiers the parent is authorised to spawn), and current Conformance certification status. Registration is at class granularity, not per instance, so, an FIU may run many instances of a registered class, but the class is the unit of identification.

Manifest uniqueness is enforced by hash so a TSP deploying the same underlying agent for multiple FIUs cannot disguise it as different agents. A single TSP agent class may serve many FIUs by listing them all in served principals; each FIU counter signs independently.

b) The Identity Layers

Three identity layers resolve at every API call: the agent class (what software is acting, resolved against the manifest), the operating pair (who runs it and on whose behalf, resolved as an operator plus served entity against the Central Registry and the manifest's served entities list), and the delegating Customer (on whose behalf the consent was granted, bound at consent grant time and enumerated in the consent handle's registered_agent_classes list).

In Pattern A, the FIU is both the operator and served entity. In Pattern B, they are distinct: the TSP is the operator and the FIU is the served entity, and every system receipt records both. If any layer fails to resolve, the receipt is structurally invalid. In Pattern C, the FIU or TSP operates the stack similar to the Pattern A or B model, however, it includes an accredited AI model inside TEE.

c) The Signing Layers

The framework operates on two signing layers.

At registration time, the operator signs the agent class manifest and the named served entities counter-sign their bindings; these are one-time signatures captured in the ACR, with the FIU retaining the right to revoke its counter-sign or the operator to deregister entirely.

At runtime, the TEE signs every receipt at the moment of action, referencing the registered manifest by identifier rather than collecting a new signature from the operator or the FIU for each receipt.

This separation is what lets the framework operate at ecosystem scale: the FIU's accountability is established once, at registration, by a cryptographic counter-sign that the regulator can audit; the per-action receipt-emission is automated by the TEE under its enclave-bound key.

d) Operator Patterns

Three operator patterns cover the deployment landscape. The patterns share a fundamental architectural property: in all three, the agent runs inside the TEE, the TEE emits and signs receipt, and the FIU is always primarily accountable to the regulator regardless of who operates the agent stack.

What differs across the patterns is the registration shape of the agent class, specifically, who signs as operator and who counter-signs as served principal at registration time and whether model inference runs outside the TEE on network-reachable infrastructure or inside the TEE under the Confidential AI extension.

A detailed description along with the differences in the patterns is provided in the table below:

Table 2: Comparative Matrix of The Three Operator Patterns

Feature	Pattern A	Pattern B	Pattern C
Operating Agent	The FIU operates its own agent stack.	Accredited TSP operates the agent on behalf of the FIU. However, the FIU remains as the principal agent.	The FIU or TSP operates the stack similar to the Pattern A or B model, but this Pattern also includes an accredited AI model inside the TEE.
Signatory	Pattern A relies on a single party for signature. The FIU signs as both operator and served principal.	Pattern B follows a dual party signature system. Here, the TSP signs as operator and the FIU counter-signs as served principal.	For Pattern C, the signatories are the same as Pattern A and B. However, it relies on who is operating the system.
Actors in the Data Path	The FIU is the sole actor in Pattern A.	The FIU along with the accredited TSP work in Pattern B.	The FIU or FIU and TSP are the central actors here as well, with provisions of using internal model infrastructure in the path.
Receipt Generation	The FIU appears in both operator and served principal fields.	In this model, since the TSP and FIU are both involved and work as the operator and served principal respectively, both are named distinctly in every receipt.	It follows a similar model as Pattern A or B. However, the model attestation field carries weight hash instead of provider identity.

Feature	Pattern A	Pattern B	Pattern C
Model Inference Location	Pattern A runs outside the TEE on FIU's own infrastructure.	Pattern B runs outside the TEE on TSP's accredited infrastructure.	Pattern C runs inside the TEE, wherein model weights are loaded into the enclave at inference time.
Data Boundary & Governance	The data exits the TEE cleartext to reach the model and is governed by PAC and manifest.	The same boundary rules as Pattern A apply with the addition that TSP's MCP server must be declared in manifest.	In Pattern C, neither AA nor model weights cross any boundary and the TEE attestation covers both simultaneously.
Model Accreditation & Oversight	Pattern A is guided as per the existing FIU oversight regulations.	Pattern B depends on the TSP's own accreditation.	Model identity verified cryptographically via weight hash and if there is a weight mismatch, the TEE refuses invocation.
Raw Data Accessibility	Raw data stays inside the TEE, so only permitted derivatives or redacted prompts may cross to the model.	Raw data never travels to the TSP, so the TSP receives only system receipt, customer receipt, and authorised output.	Raw data never leaves the enclave, so the model sees it only inside the TEE boundary.
Exfiltration Risk Control	The TEE enforces PAC and manifest. Therefore, all model calls run through the registered MCP path.	TEE access count per consent handle in every receipt gives FIU continuous visibility into what TSP's agent can read.	Exfiltration risk is structurally eliminated by restricting any outbound model call.

Feature	Pattern A	Pattern B	Pattern C
Open-Source Model Verifiability	Pattern A has a standard system without any specialised mechanism.	Pattern B also has a standard system without any specialised mechanism.	For Pattern C, an industry body will have to confirm the manifest declaration.
Use of Confidential AI	FUIs declare model hash in manifest and the TEE verifies at inference time.	Similar to Pattern A, Pattern B follows the same mechanism.	Confidential AI is not an extension but the defining property of Pattern C.
Proprietary Model Support	The FIU can run its own model outside the TEE and in this, the weights are not protected from FIU's own infrastructure operators.	TSP can run its own model wherein, the weights will be visible to the TSP's infrastructure operators.	Weights are loaded into the enclave. But the TEE operator sees the measurement (hash), not the weights, thereby protecting proprietary fine-tunes.

* Please note: FIU is always primarily accountable to the regulator regardless of who operates the agent stack.

d) MCP Servers and the Agent Tool Surface

Model Context Protocol (MCP) is rapidly becoming the standard way agents call tools. Because every agent runs inside the TEE, the TEE attaches an MCP wire observer to every agent and derives every receipt from observation of the protocol wire between the agent and its tools, a vantage point the agent cannot signal to or suppress.

Registration: MCP servers used by registered agent classes are themselves registered in the Agent Class Registry. Each registration carries a capability manifest: the tools the server exposes, their side effect classification, their sensitivity classes, and a server image hash.

Observation: Every MCP invocation by a registered agent class produces an entry in the system receipt for that action, referencing the registered server identifier, the specific tool invoked and the tool's side effect classification. Because every agent runs inside the TEE, the TEE derives the receipt entry from the MCP wire observer attached to the agent, not from the agent's Software Development Kit (SDK) callbacks, because the wire observer sees the actual

data that crossed the protocol.

Enforcement: Because every agent runs inside the TEE, the control point is the same point at which the TEE already enforces consent and class permissions, so enforcing mode is available to any class, not only consequential ones. It is the default wherever the constraint guards a consequential action, and is available as a declared option for derive only and internal action classes whose composition the consent chooses to constrain.

3.1.3 Provenance

Provenance largely deals with the boundary of the data, which elucidates which actions can be taken under the framework and which actions violate the jurisdiction. In this, the receipts and Processing Receipt Registry become the gateway for oversight.

Receipt Generation

Each agentic action triggers a system receipt from the TEE, which records what was read, by which agent class, under which consent, the tool calls by argument and result hash, and the output classification. System receipts chain back to the read receipt, so any action can be traced to a confirmed access of data the FIP placed and the consent authorised. Where the TEE blocks an action before it runs, it emits a refusal receipt in place of the system receipt that action would have produced, anchored into the same receipt tree, so a blocked action is accounted for rather than absent. Similarly, three further receipts close out the lifecycle of a consent rather than record a live action: a termination receipt, emitted when an in flight invocation is wound down because the consent was revoked mid workflow; an expiry receipt, generated when the consent reaches the end of its declared data life; and a deletion receipt, generated as proof when a dataset or derivative is disposed of under the retention rules. These lifecycle receipts let the coverage diff resolve a dataset that was ended rather than processed, so a consent that was revoked or expired closes cleanly instead of leaving an apparent gap.

a) Types of Receipts

There are four operational receipt types in the framework, plus one rights artefact derived from them.

- **Delivery Receipt**

The TEE emits a delivery receipt on FIP placement. The receipt carries the hash of the placed dataset and the data reference that the AA will pass to the FIU, which is then anchored to the data fetch identifier the AA has assigned. It carries the hash of the decrypted dataset so the FIU can verify on decryption that its retention copy matches the dataset the TEE processed.

- **Read Receipt**

The TEE issues a read receipt when the FIU first accesses the data through the reference, verifying that the dataset matches the hash on the delivery receipt. Thus, this receipt is the gateway that introduces the consent handle into the chain. It verifies the dataset present at the reference, matches the hash declared in the delivery receipt, and establishes the consent context for downstream system receipts.

- **System Receipt**

The system receipt is emitted by the TEE for every agentic action under a consent, anchored to the consent handle. This receipt records what was read, by which agent class, the tool calls by argument and result hash, and the output classification. It is then chained to the read receipt that preceded it.

- **Refusal Receipt**

Where the TEE blocks an action before it runs, it emits a refusal receipt in place of the system receipt that action would have produced, anchored into the same receipt tree, so a blocked action is accounted for rather than absent. A refusal receipt is a signed record that an action was attempted and blocked.

The refusal receipt closes a gap that a purely Processing Receipt Registry-side check would leave open: if the Processing Receipt Registry merely rejected a non-conforming receipt at ingestion, the action it described would already have happened, and its rejection would leave no record at all. The Processing Receipt Registry-side check remains, but as secondary verification that what arrived is consistent with what the TEE should have permitted, not as the primary control. The framework enforces before the action and verifies after the receipt.

- **Customer Receipt**

The customer receipt is the rights artefact derived from the above four receipts. The Customer sees a customer receipt for each consent: a plain-language summary of what processing occurred. The Customer views it through the AA, the same consent dashboard where they granted the consent and where they can revoke it. The connection runs through the consent handle: the AA already holds the mapping from Customer to their consent handles (it issued the consents), and for each handle it requests the customer receipt the Processing Receipt Registry has composed from that consent's system receipts.

The Processing Receipt Registry is queried by consent handle, never by consumer identity, and releases a consent's customer receipt only to the AA that issued that consent, so consumer-identity resolution stays at the AA and the Processing Receipt Registry stays a handle-keyed Registry. Because the customer receipt is a data-free summary (descriptions of activities, not the underlying financial data), surfacing it through the AA does not disturb the AA's data-blind role.

b) Receipt Integrity

None of the receipts hold data but they hold references to the data the TEE holds for the duration of the consent and to the derivatives the FIU receives back. Receipt integrity is uniform because all agentic processing on AA data runs inside the TEE, with no exceptions. This includes derive only processing: reading and deriving from AA data is exactly the opaque activity the framework exists to make observable, so a derive only agent is no more exempt from the TEE than a decision making one.

c) Enforcement

Enforcement happens before the action runs, not after the receipt arrives. This distinction matters. The PAC block states which agent classes and processing classes the consent permits. The TEE obtains the signed consent artefact that carries the PAC block from the AA, the consent manager that issued it and passed it to the TEE as part of the same dispatch that routes the data.

The artefact is signed, and for consequential decisions carries the Customer's WebAuthn assertion bound to the consent payload, so the TEE enforces against a signed PAC block it verifies rather than an asserted one. Because all agentic processing of AA data runs inside the TEE, the TEE holds the consent artefact with its PAC block, resolves the agent class to its ACR manifest, and checks before allowing the agent to act that the class and the processing it is about to perform are within what the consent permits. If they are not, the TEE refuses the action before any tool call is made or any output produced, and emits a refusal receipt in its place.

Enforcing at the TEE, rather than rejecting at the Processing Receipt Registry, is the difference between preventing an unaccounted action and discovering its absence too late to act.

Processing Receipt Registry (PRR)

a) Coverage

Coverage of the consent is a property the Processing Receipt Registry can compute, not a property of any single receipt. For every active consent, the Processing Receipt Registry keeps a record of when the data was first accessed and maintains the hashes of all source artefacts referenced by the system receipts generated under that consent. Together, these records provide a verifiable audit trail of processing activity.

b) The Processing Receipt Registry

The Processing Receipt Registry holds no personal data. Receipts are keyed by the consent handle, which is an opaque pseudonymous token; the mapping from a consent handle to an identified Customer lives only at the AA that issued it, so the Processing Receipt Registry cannot resolve a handle to a person on its own. Receipt contents are hashes, references, and classification labels rather than raw financial data or direct identifiers. The Processing Receipt Registry itself runs inside a TEE, so even its operator cannot read or alter the Registry at runtime, and queries against it are served from within the enclave. The Processing Receipt Registry is a pseudonymous Registry, not a consumer-data store, which is the same handle-based separation the existing AA framework already relies on. Figure 1 shows how the operational receipts compose into an end-to-end chain from FIP placement to FIU action.

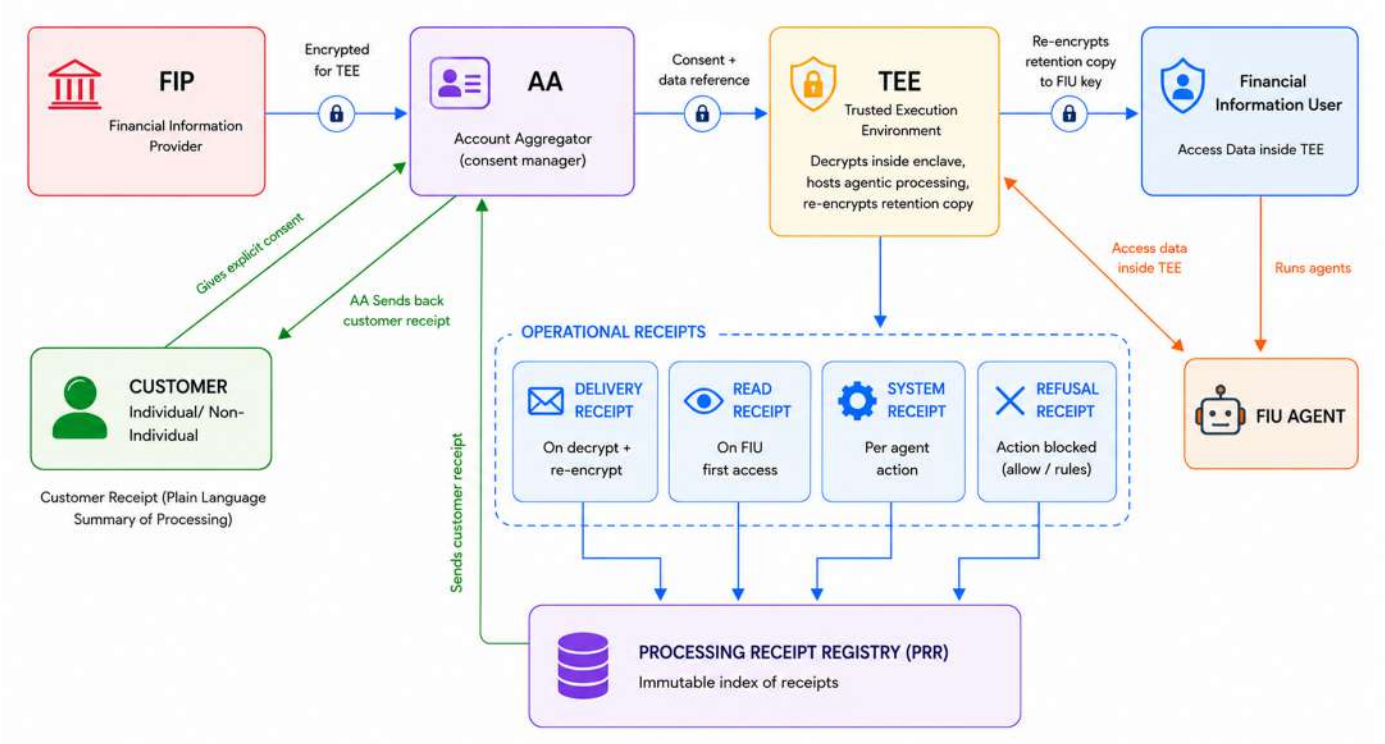


Figure 1. End-to-end receipt flow from FIP to FIU. The FIP encrypts the dataset to the TEE's enclave-bound public key and dispatches via the AA. The TEE decrypts inside the enclave, hosts all agentic processing on its copy, and re-encrypts a retention copy to the FIU's key. The TEE emits a delivery receipt on the FIP's behalf, covering decryption and re-encryption; the AA passes the data reference to the FIU; the TEE emits a read receipt on the FIU's first agentic access through the reference; system receipts follow each agentic action, all under TEE wire observation. If the TEE blocks an action before it runs (for example, because the consent does not permit the class or the processing, or the consent is revoked), it emits a refusal receipt in place of the action, so a blocked action is accounted for rather than leaving a silent gap. Cleartext exists only inside the enclave; the FIU's retention copy is encrypted by the TEE under attestation.

3.1.4 Conformance

The fourth pillar establishes whether an agent class behaves within its declared manifest under realistic stress, before it acts on AA data in production.

The industry body maintains the **Conformance Library:** a versioned collection of test cases covering schema, behaviour under revocation and scope changes, robustness, and operational practices. New attack classes enter through ecosystem review and publish as versioned releases that trigger re-certification.

The industry body certifies agent classes against the Conformance Library; classes that pass are recorded in the ACR with their certification status.

An agent class must pass the current library version before production. A class that commits to a consequential decision faces harder testing than one that only produces a derivative. Each manifest in the ACR carries the current attestation status, allowing the Processing Receipt Registry to validate every system receipt against it, so a lapsed attestation is detectable at runtime and audit.

Model identity is bound differently depending on where the model runs.

When inference happens outside the TEE as in Patterns A & B, the manifest declares the model and the framework relies on contractual commitment, supported by provider response metadata and the FIU's audited deployment. Behavioural conformance tests the declared model, so substitution would create divergence over time, but the substitution itself is not directly detectable from receipts.

Under Confidential AI in Pattern C, the model runs inside an attested TEE and its weight hash is part of the measured boot. The TEE refuses invocation if the loaded model weights do not match the manifest. Each receipt cryptographically binds the decision to the specific model-weight hash used at runtime.

For published open-weights models, any third party can independently download the released weights, compute the hash, and confirm that it matches the manifest. Standard deployment patterns provide contractual model identity; Confidential AI provides cryptographic model identity.

3.2 Parent and Child Agents

Most consequential agentic workflows are not single agents. They are compositions. A parent agent orchestrates the workflow, holds the consumer's top level consent, and supplies the strategy. Child agents perform specialised sub tasks under delegated authority. Children may themselves spawn grandchildren. The result is a tree of agent invocations, each with its own scope, its own tool surface, and its own audit trail. The underwriting workflow in Section 4 is the canonical example: a parent orchestrator with five children, none of which sees more than it needs to do its job.

Parent and child describe a role within a specific workflow, not a property of the agent class itself. The same class may run as a parent in one workflow (an orchestrator coordinating its children) and as a child in another (the same orchestrator class spawned by a higher level decision engine). The class is what the agent is; parent or child is what it is doing right now.
Finally, let's look at two structural rules.

First, a parent can only spawn children whose class identifiers appear in its own manifest's permitted_children list. The runtime cannot manufacture new children at runtime; only manifests already registered with the industry body can participate, and the parent must have pre-declared which classes it may delegate to.

Second, a child cannot exceed the scope granted to its parent; scope reduces as the workflow descends, never expands. Each agent in the tree has its own manifest, its own Conformance certification, and its own system receipts emitted by the TEE for each action it takes. A parent's receipts reference the consent handle; a child's receipts reference the parent's system receipt for the spawn action, which creates the verifiable chain. Parents are not absolved when children misbehave, and children are not absolved when parents over-authorise. Both layers are independently auditable.

4. Architecture

In the previous section, what the framework entails has been clearly mapped. However, it would be worthwhile to study the entire structure in order to gain a holistic understanding of how the system operates. Thus, this section attempts to provide a birds-eye view of the architectural structure of the operation of the AI agents in the AA Ecosystem.

Drawing from the original model, the Customer continues to enjoy the central position in this architecture as well. Each step is dictated by the consent of the Customer and Customers can revoke their consent at any step of the way. As soon as the consent is revoked, the TEE is

alerted and the processing of any data is stopped immediately, resulting in a refusal receipt being generated for anyone who seeks to use the data.

To put things into perspective, imagine a Customer reaches out to an FIU with a lending request. The FIU has to connect with the AA and ask for the necessary information to process the Customer's request. The AA, in turn, seeks consent from the Customer to authorise the FIP to share the information with the FIU. If the AA receives consent, they direct the FIPs to share the information as a processed file to the FIU within a safe environment, which in this case is the TEE. Here, the TEE acts as a secure network where all data processing and FIP-FIU interactions take place.

Within the TEE, the FIU may access and process the consented data in accordance with the permissions granted by the customer and the applicable policy controls. The raw data remains protected from the infrastructure operator and any party outside the TEE, while all processing activities are governed by attested policies and recorded through processing receipts.

Once the data enters the TEE, a processing receipt is generated at every step. Be it a read receipt, a delivery receipt, a system receipt or a refusal receipt, each step leaves behind an audit trail, clearly indicating what action has been taken and what was the consequence of the action. Here, a customer receipt is also generated to ensure that the customer remains in the loop. However, even before receipt generation takes place, the TEE is tightly secured by an industry body, which verifies all the systems acting within the TEE. This authorisation stems from the ACR, which is like a repository of all agents functioning in the TEE. If an agent is not registered in the ACR, they are forbidden to operate in the TEE.

This includes the declaration and registration of any AI-enabled system as well. This means that an FIU, which may wish to use an AI agent in its processing, must disclose the AI agent being deployed, its scope, limitations as well as its accreditation. On one level, this ensures easy traceability, and on the other, it acts as a safeguard against the incorporation of malicious or unauthorised agents into the TEE. The ACR is further complemented by the PAC block which clearly outlines the extent to which the data can be used by the FIU. This keeps a check on the parent-child agents and ensures that the data is not spawned or harnessed beyond its jurisdiction, while also delineating which derivatives of the data may be retained and for how long. Alongside, the industry body looks into the Conformance Library to verify that the agent works within its declared manifest.

Here, the industry body provides an oversight mechanism for the TEE and meticulously verifies every agent that enters the TEE. So the approved agents, including the AI systems function within a defined boundary. Once this boundary is established, the data processing by the FIU

takes place within the TEE. This eliminates the risk of identity theft, financial fraud or simply the use of financial data for AI training and creates a secure space for data processing. Thereafter, the FIU can use the processed data to honour or reject the Customer's request.

To put it simply, an illustration of the above architecture is also given below:

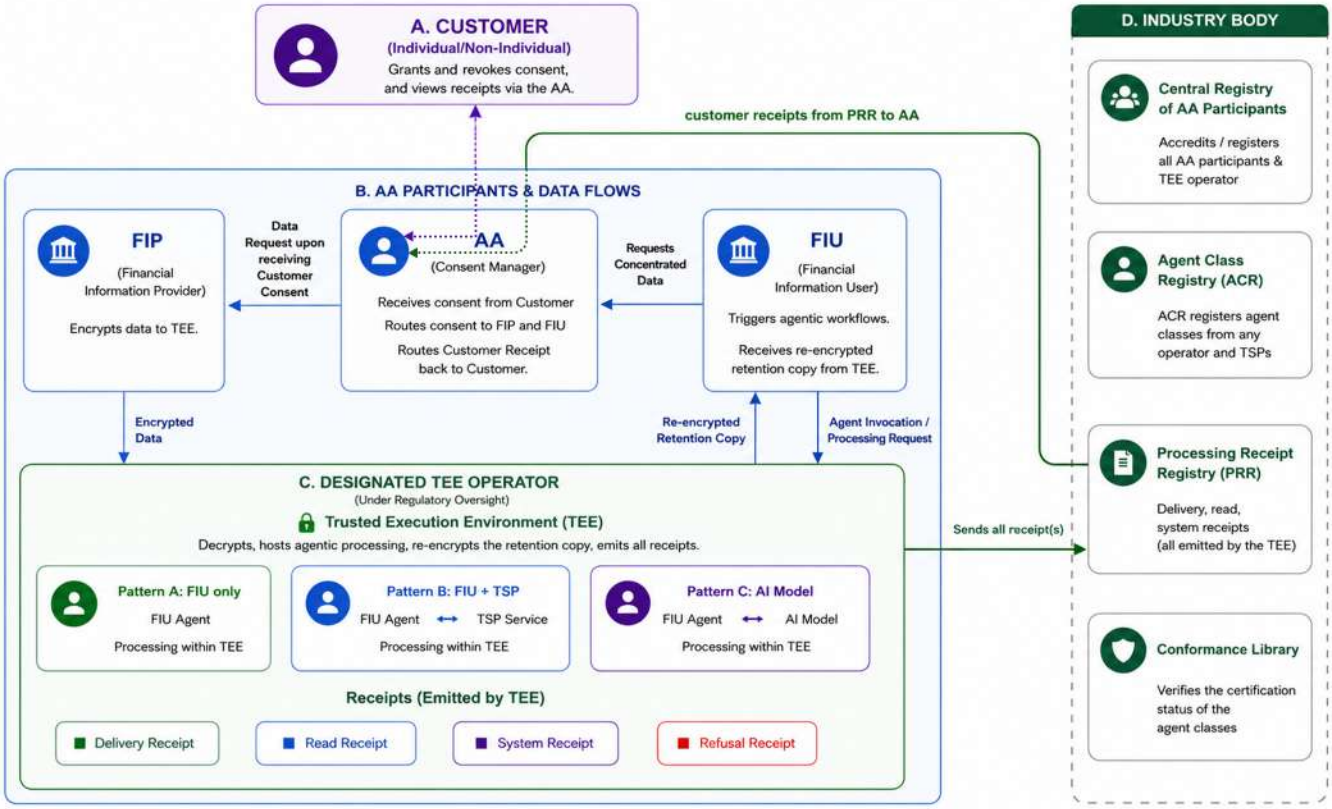


Figure 2. Architecture showing how industry-body operated infrastructure, AA participants, and the operator (in any of its three patterns) connect. Registration and accreditation flows are dashed; receipts and operational flows are solid.

4.1 Trusted Execution Environment (TEE)

The framework mandates an entity that holds the Customer's financial data in cleartext, inside the enclave, between the FIP's dispatch and the FIU's receipt of its retention copy. Therefore, the TEE acts as a Data Processor on behalf of the FIU. The FIU determines the purpose and means of processing. It deploys the agent classes, holds the consent relationship with the Customer, and is the accountable party for the decision, whereas the TEE operator provides the protected runtime in which that processing is executed and observed under the consent obtained by the FIU. This places the TEE in the same AA regulatory posture as any processor an FIU engages to handle consented data, with the distinction that the TEE is independently designated, accredited, and supervised, and that its operations are constrained by attestation rather than only by contract: the TEE cannot interact with cleartext at runtime, so its processor

role is bounded by the enclave architecture and not merely by its processing agreement.

Two consequences follow. First, the FIU's processing agreement with the TEE operator is the instrument that establishes the processor relationship, and the framework expects this agreement to exist as a registration prerequisite, alongside the operator's accreditation. Second, the operator's accreditation and supervision prevent the processor relationship from becoming a route around the FIU's obligations. Because the operator is supervised independently and emits receipts that the FIU cannot alter, the FIU cannot use the processor relationship to place processing beyond its own accountability. The receipts make the FIU's processing visible to the regulator precisely because the operator, not the FIU, emits them.

The framework treats the operator as a processor rather than as a separate or joint entity deliberately. A joint reading with the FIU would fragment accountability for the decision across two parties and weaken the single line of responsibility the AA framework draws to the FIU; the framework's position is that the FIU remains singularly accountable for what is done with the Customer's data, and the TEE operator is the supervised infrastructure through which that processing is made observable, not a co-owner of the decision.

4.1.1 Attestation in the Flow

When a TEE enclave boots, the confidential-computing hardware measures everything loaded into the enclave (code, configuration, and model weights in case of Confidential AI) and signs that measurement with a silicon-rooted key. The TEE operator presents this attestation report to the industry body, which verifies it through the silicon vendor's verification service and, if the measurement matches an approved value, publishes the corresponding enclave public key as active in the registry.

When the FIP dispatches a ciphertext, its client library fetches the current attestation report for the target enclave key, verifies it against the silicon vendor and the industry body's published valid set, and proceeds only if all checks pass. A failed verification stops the fetch; the FIP does not encrypt to an unverified key. This is the fail-closed property the framework relies on at ingest.

At runtime, every receipt the TEE emits is signed by the enclave-bound key tied to the attestation, so any reader can verify the receipt's provenance back to an attested enclave. At audit, the industry body retains the history of valid measurements and public keys with their validity windows, so a receipt from years ago is verifiable against the attestation that was current when the action took place, not only the current one. Under Confidential AI, the same machinery additionally verifies model weight hashes through the model TEE's attestation chain.

4.1.2 Operation of the TEE

The framework's position is that the TEE is run by a separately constituted, independent operator, set up specifically to operate the protected runtime under regulatory oversight. The operator must be neutral by design. The framework does not pre-commit to the identity of that operator beyond these requirements. What matters is that the operator is independent of the TSPs and FIUs and is accredited, supervised, and subject to incident disclosure and audit by the regulator.

The framework's integrity properties are structural: every action produces a receipt that only the enclave can sign, the consent's PAC block is enforced before any action runs, AA-fetched cleartext is confined to the enclave with the TEE operator unable to read it at runtime, and the enclave-bound keypair ties access to the attested measurement. These properties hold for any operator that is accredited, and supervised.

The industry body's core role in the framework is the governance layer: it operates the Agent Class Registry, the Processing Receipt Registry, the Conformance Library, and it sets the rules these encode.

Whoever is designated to operate the TEE, that separation between ecosystem rule-setting and runtime infrastructure operation should be maintained, with both answerable to regulatory bodies.

4.1.3 FIU-Operated TEE Nodes

An FIU may operate a node of the framework's TEE network on its own confidential-computing hardware, accredited separately as a TEE operator from its accreditation as an FIU. This is a multi-operator deployment of the same shared infrastructure, not an exception to it. The integrity properties the framework relies on (uniform measured code, common attestation chain, receipts to a shared Processing Receipt Registry) are properties of the network, not of any node's location. They hold whether the operator is a separately designated entity or an accredited FIU operating in its operator capacity.

The conditions are that the node runs the framework's published enclave code with measurements chaining to the industry body's published valid set, attests through the same silicon-vendor verification service every other node uses, and emits receipts into the Processing Receipt Registry without local mediation. The FIU cannot suppress receipts about its own actions; the attestation enforces this rather than the operator promising it. The FIU's operator role is audited separately from its FIU role, with controls preventing the same individuals from

controlling both day-to-day. Multi-region failover routes traffic elsewhere if the node is unavailable.

The framework does not accept FIU-internal TEEs running FIU-internal software as equivalent infrastructure. A private enclave running private software cannot produce receipts the network can validate, or be tested against the Conformance Library every other class is certified against. The framework's commitments attach only to processing that runs on accredited nodes of the federated network.

4.1.4 Runtime, Enclave, and Invocation

The TEE is a runtime, a service operated by the designated TEE operator within which long-lived enclave instances handle dispatch from FIPs, decryption of consented data, agent orchestration, consent enforcement, receipt emission, and re-encryption of the retention copy to the FIU.

The enclave is persistent and attested for the duration of its validity window; the invocation is the per-action execution scope inside it. The framework's integrity properties are enforced at the enclave boundary and automatically apply to all processing performed within the enclave. These properties include the emission of TEE-signed receipts for every action, protection of cleartext data from the operator at runtime, and pre-execution enforcement of the consent's PAC block.

The runtime scales horizontally across multiple enclave instances within India, with the FIP's client library routing dispatch to whichever enclave is healthy.

4.2 A Case Study: Agentic Underwriting

To develop a concrete understanding of the four pillars and the architecture, consider an FIU running a lending agent on a personal loan application for INR 25 lakh. The lender deploys a parent orchestrator and five children. Three FIPs hold the Customer's bank accounts (a primary bank, a savings bank, and a salary account bank); all that the FIU receives via AA is the bank statement data from these accounts. Every signal the children produce, income, expenses, existing debt obligations, is derived from those statements.

Table 3: Agent Classes and Scope for the Case Study

Agent Class	Scope
Underwriting Orchestrator (parent)	Holds the consent. Spawns children. Reads raw data directly.
Income Assessment	Reads 24 months of bank statement credit transactions. Identifies recurring credits versus one off transfers.
Liability Estimation	Reads 24 months of bank statement debit transactions. Categorises spending.
Credit Assessment	Reads bank statement debit transactions. Detects recurring EMI shaped outflows; infers debt service burden.
Debt Serviceability	Analyzes financial statements to check debt repayment ability.
Financial Discipline	Checks for payment defaults.

*In this model, only five agent classes have been shown, however, if required, more agent classes can be added in the model.

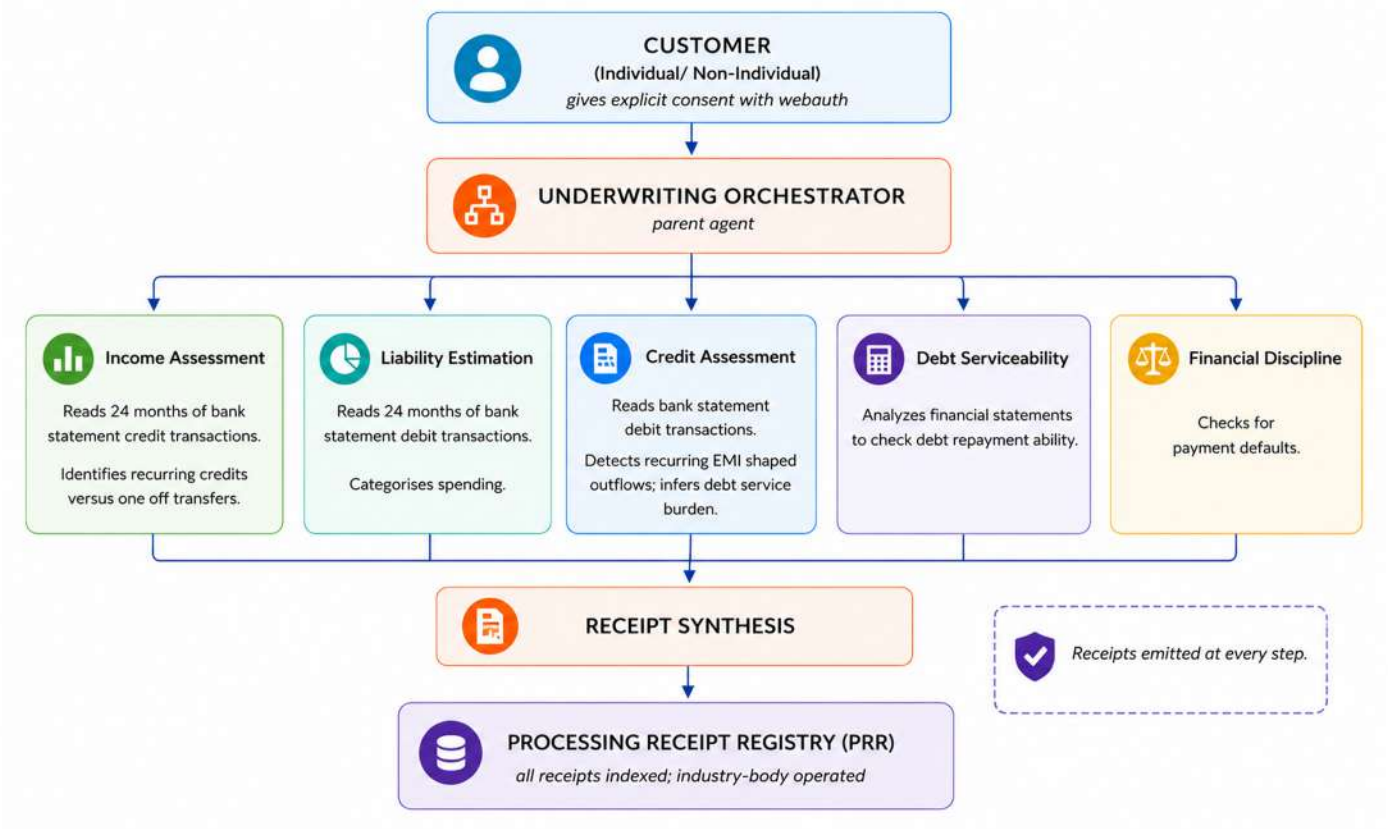


Figure 3. End-to-end Underwriting Workflow.

In flow: the Customer signs the consent in the lender's app with WebAuthn. Identity resolves across the three layers (agent class, operator, Customer). Before any child acts, six independently auditable checks fire: the consent is active and its PAC block authorises this processing; the child class is registered in the ACR; the child is in the parent's permitted_ children list; the child holds a current Conformance certification; the child's declared scope is a subset of the parent's; and the WebAuthn assertion binds the consent the workflow is running under. Data fetches through AA proceed as today, with the one change established earlier that the recipient is the TEE rather than the FIU. Each child operates under its own manifest with a strict subset of the parent's scope. Every agent runs inside the TEE, so signed receipts emit at every step under TEE wire observation, which the FIU cannot suppress without breaking its declared topology. If the Customer revokes the consent mid workflow, the TEE refuses new invocations and in flight invocations complete with termination receipts.

The four pillars compose into end to end accountability without the industry body operating anything inside the FIU's perimeter: the agents run in the TEE, while a separately designated entity operates outside every participant's perimeter, and the industry body operates only the governance layer that holds the registry, the Processing Receipt Registry, and the conformance Library.

4.3 Benefits for FIUs

The framework constrains FIUs in specific ways: receipts that cannot be suppressed, manifests registered before classes act, attestations that bind decisions to specific code and weights. They are the infrastructure, FIUs benefit from using in five areas.

- **Defensible compliance**

When a Customer disputes an agentic decision today, the FIU defends itself by reconstructing what happened from internal logs that are partial, mutable, and self-attested. The defence rests on being trusted. Under the framework, the receipt chain, manifest and attestation gives the FIU positive evidence of compliance, signed by infrastructure the FIU does not control and therefore credible to a regulator and a court. The same property that constrains the FIU is what makes the FIU's compliance defensible.

- **TSP risk reduction**

An FIU using a TSP today extends data trust to an entity outside its control while remaining accountable as per applicable regulatory and legal requirements. Under the framework, the TSP's agent runs inside the shared TEE infrastructure; raw AA data never reaches the TSP's infrastructure. The FIU gets to use specialised TSPs without bearing the exfiltration risk today's arrangements carry.

- **Shared governance infrastructure**

Every FIU deploying agentic processing today builds its own audit logging, consent enforcement, model versioning, and decision traceability. The framework provides these once, at ecosystem level. The FIU registers classes, passes conformance, and deploys; it does not maintain a parallel governance stack per product.

- **Faster regulatory approval**

A regulator engaging an FIU's new agentic product today has to take the FIU's word for what the agent will and will not do. Under the framework the regulator verifies the manifest, conformance status, and behaviour directly against shared infrastructure both parties trust.

- **Cross-product consistency**

Lending, fraud, KYC, and insurance products today carry separate governance per product. Under the framework all of them register through the same ACR, run inside the same TEE, and emit to the same Processing Receipt Registry.

5. Operating the Framework

The framework requires two infrastructure roles: a Governance Operator and a Runtime Operator.

An industry body holds the governance layer: it operates the Agent Class Registry, the Processing Receipt Registry, and the Conformance Library, and it sets the rules these encode.

A separate designated entity, operating under regulatory oversight, runs the TEE (Runtime) that holds and processes AA data. Keeping these roles distinct keeps the rule-setter and the runtime operator separate; no party holds both ecosystem policy authority and runtime control of the data path.

The industry body's governance role covers the TEE infrastructure as well as the agent layer. A TEE node's standing in the federated network is conditional on continuous certification across three dimensions. The enclave's measured boot must match a measurement on the industry body's published valid set; modified code produces a different measurement and is refused admission. The underlying confidential-computing platform must be on the framework's accepted hardware list, with a firmware version the industry body accepts, so deprecated platforms or vulnerable firmware lose standing. And the entity running the node must be accredited as a TEE operator, with audited separation from any other ecosystem role it holds.

Standing is continuous, not one-time. A node falling out of certification on any dimension drains

its in-flight load and stops accepting new traffic; the validity-window machinery (Section 2.2) handles the transition without disrupting the ecosystem.

The Processing Receipt Registry is the live Registry of what agents did; the Library is the test bench that agent classes pass before being certified for production. The two connect in the ACR: a class's Library pass status is what the Processing Receipt Registry validates each system receipt against. Routine work on the governance side (manifest validation, certification scheduling, Processing Receipt Registry integrity) is handled by an AI agent the industry body builds and runs against the same Processing Receipt model the framework asks of FIUs. Humans supervise policy decisions: new attack classes, contested attestations, deregistration.

The framework's default posture is Human-On-The-Loop for processing and Human-In-The-Loop for governance. Receipts, audit, and grievance let humans inspect, dispute, and intervene after the fact, with structural enforcement (TEE-bound consent, conformance certification, refusal receipts) preventing entire classes of bad action before they can occur. Per-decision human review for routine processing does not scale and degrades to rubber-stamping at ecosystem volume; the framework concentrates human attention where it does real work: setting policy, vetting new agent classes, reviewing the conformance Library, responding to grievances.

Within processing, Human-In-The-Loop remains available as a declared property of the consent's PAC block, suitable for new-class probation windows the agent itself escalates. The choice between in-the-loop and on-the-loop for a given consent is the Customer's to make at consent grant time, within a floor the regulator sets and the constraints the FIU declares, not a runtime improvisation.

The TEE is what makes the framework's agentic-processing observability enforceable rather than policy bound. It is shared infrastructure, operated by a designated entity under regulatory oversight, that receives each AA-fetched dataset encrypted to its enclave-bound public key. Inside the enclave, the TEE decrypts the dataset, hosts agentic processing on the cleartext, and re-encrypts a retention copy to the FIU's key for the FIU to retain as per applicable regulatory and legal requirements. Three properties matter for the TEE's role. Cleartext lives only inside the enclave, not at the FIU's perimeter and not on the operator's systems. Because the TEE is the emitter of all delivery, read, and system receipts, neither the FIU nor a TSP can suppress or modify what the receipt says about what an agent did. Registered agent classes execute inside the TEE, not against a remote copy of the data, which means agentic processing produces only receipts and the derivatives the consent authorised.

All agentic processing on AA data runs inside the TEE, without exception, including derive only processing. This is a registration prerequisite for every agent class, not only those producing

consequential decisions: deriving from AA data is itself the opaque activity the framework exists to make observable, so it is not exempt. The rule extends to TSPs operating under Pattern B: a TSP's agent runs inside the TEE ecosystem on submission, never on the TSP's own infrastructure, so raw AA data never reaches the TSP (the FIU's retention copy is the FIU's own; the TEE copy is the TEE's). The exfiltration risk where a TSP would otherwise retain raw FI data and build parallel products from it is closed structurally, because the data the TSP would copy is never on the TSP's side to begin with. Running every agent inside the TEE, rather than letting low stakes jobs run in an FIU's own perimeter, is a deliberate choice: an outside path would reintroduce both the opacity the framework removes and an unenforceable revocation, and the shared TEE is sized to carry the load.

The Conformance Library is a living artefact the industry body maintains, drawing on the SAFE MCP, OWASP LLM Top 10, and MITRE ATLAS. It tests schema and format, behaviour under revocation and scope changes, resistance to prompt injection, and tool-chain abuse, and operational practices like key management and sub-processor disclosure. The full breakdown is in the glossary entry. New attack classes enter through ecosystem review by the industry body, are vetted for reproducibility and severity, and are published as versioned releases that trigger re-certification for affected classes. The Library runs inside TEE-backed environments so FIUs and TSPs do not have to expose proprietary model weights or prompts; only the pass or fail result emerges. Failure records are sticky: a participant cannot shop for a favourable second pass while a failure stands on record.

5.1 Sandbox for Experimentation

The framework adopts a permissive approach to innovation by providing a sandbox where FIUs and TSPs can develop, test, and evaluate new agent classes before they are allowed to operate on live AA data.

Sandbox environments are restricted to synthetic or anonymised data and do not have access to production consents or live AA fetches. Sandbox agents are registered separately from production agents but operate under the same TEE-backed runtime and receipt model, allowing their behaviour to be observed under realistic conditions.

Promotion to production requires Conformance certification appropriate to the agent's permitted actions.

6. Audit and Grievance

A Customer exercising grievance rights starts from the customer receipt for the consent, which summarises in plain language which agents acted and what processing occurred. The Customer presents the consent identifier and the decision in dispute. The FIU's grievance officer accesses the corresponding system receipts in the Processing Receipt Registry and follows their references to the FIU's decision dossier, the structured record of prompts, tool calls, policy checks, and intermediate outputs that the TEE returns to the FIU at end of processing, alongside the retention copy. The dossier sits inside the FIU's perimeter because the FIU, as the accountable party, holds the supporting evidence for decisions made on its behalf; the PRR receipts make the dossier addressable and verifiable from outside.

A regulator investigating a complaint takes a parallel path: resolve the Customer's identity to consent handles at the issuing AA, query the PRR for system receipts on those handles, reconstruct the receipt tree from parent spawn to final action, and verify each manifest and the attestation current at the time.

Comparing what happened (receipts) against what was authorised (consent) and what is permitted (manifest and attestation) produces structured evidence rather than a reconstructed narrative. Where an action was attempted and blocked, a refusal receipt sits in the tree in its place, so a missing node always means a missing action.

For the FIU's own internal audit, the receipt chain produces what reconciliation across disconnected logs produces today, but structurally and at the PRR rather than through internal effort.

7. Closing

A reasonable question: does this framework move Open Finance towards 'Closed Finance'? It does not. The consumer's right to authorise any accredited participant to receive their data is untouched.

What the framework constrains is where agentic processing of that data runs inside an accredited TEE rather than wherever an FIU chooses and that constraint is what makes the processing accountable.

The framework is operator-agnostic: the integrity properties (attestation, enclave-bound keys, receipts emitted by the runtime independent of the agent) hold for any qualifying operator, so the regulator may accredit as many as the ecosystem needs, the same way it accredits AAs today.

The framework is also data-category-agnostic. The same registration, attestation, and conformance machinery governs agentic processing of any data AA carries, including the commercial and operational data categories entering AA's scope through ongoing ecosystem work. The framework does not require redesign as AA's data perimeter expands.

Where more than one is accredited, the FIP encrypts each fetch to the enclave key of the operator the consent routes to, resolved through the attestation step it already performs, so the FIP's client library holds more than one set of active keys rather than changing how it dispatches. Receipts from any accredited operator federate into the same Processing Receipt Registry, so ecosystem-wide observability is preserved without requiring a single emitter.

The AA ecosystem succeeded by establishing trust through process. The Customer could trust the framework because it was verifiable: who fetched the data, on what authority, with what record. AI agents now form a new layer between consent and outcome; the same trust property has to extend to them. The four pillars proposed here are how to do it, each extending a primitive the specification already defines. WebAuthn closes the agentic re-authentication gap without disturbing the service token layer that holds the network together. The governance role widens: the Agent Class Registry, the Processing Receipt Registry, and the Conformance Library are the three pieces of governance infrastructure an industry body operates, with the sandbox running alongside as the on-ramp for everything new.

The TEE that holds and processes AA-fetched data is operational infrastructure run by a separately designated entity under regulatory oversight; who that entity is, is a regulatory decision.

The framework provides a common trust layer for agentic AI in the AA ecosystem. By establishing shared rules for consented AI processing, agent identity, provenance, and conformance, it reduces the uncertainty under which ecosystem participants and regulators currently operate.

Today, many institutions remain cautious about deploying AI agents on consumer financial data because the accountability, auditability, and regulatory scrutiny questions are unresolved. In the absence of common guardrails, each institution develops its own controls, assurance mechanisms, and governance processes, often duplicating work and producing inconsistent outcomes. The cost is borne both by the institution building the controls and by the regulator evaluating each set of controls separately.

The framework's contribution is to provide this trust infrastructure once, at the ecosystem level. Agent developers continue to build new models, workflows, products, and services; FIUs

continue to choose their technology providers and operating models within the framework's accountability surface. What changes is that AI processing on consented financial data becomes observable, attributable, and auditable through shared mechanisms rather than per-institution reconstruction. The result is that the work of deploying a new agentic product reduces from "build a complete governance stack and defend it to the regulator" to "register a class, pass conformance, deploy."

Shared trust infrastructure has historically expanded what new technologies can do, not constrained it. Digital certificates enabled e-commerce. Payment network rules enabled global card acceptance. The framework's intent is to provide the equivalent trust infrastructure for AI agents acting on consented financial data in the AA ecosystem.

The framework enables broader and faster adoption of AI by establishing the confidence required for institutions, regulators, and consumers to participate at scale.

Sahamati Labs is publishing this draft so the ecosystem can engage with it and shape it before the first regulator-reportable agentic incident makes that conversation reactive rather than considered.

8. Agents Operating Outside AA

While this framework is designed for the AA ecosystem, the governance challenges it addresses are not unique to AA. As AI agents gain direct access to consumer financial data through APIs and other bilateral arrangements, the same questions arise: what authorises an agent, who it is, what it did, and whether it operates within approved bounds.

The mechanisms used to answer these questions may differ outside AA and will depend on the institutional structures that emerge in those environments. However, the four pillars of Consent, Identity, Provenance, and Conformance provide a governance model that can extend beyond AA-mediated processing.

The framework’s contribution is not to prescribe how agentic access should be governed in every context, but to demonstrate one operational approach. As agentic systems become more prevalent, the broader challenge of governing access to consumer financial data will require collaboration across ecosystems, institutions, and jurisdictions.

Appendix A: The Shared Infrastructure Model

This appendix addresses the architectural assumption flagged at the top of the paper. It is a standalone discussion, intended to be readable on its own and to give regulators or reviewers enough to engage with the shared infrastructure choice substantively. The same content is also published as a separate one-page companion note.

The Choice

The framework introduces shared infrastructure for agentic governance: an Agent Class Registry, a Processing Receipt Registry, a Conformance Library, and a runtime TEE.

The AA primitives viz. FIPs, FIUs, AAs, and the consent flows between them remain as they are today. What changes is that the governance and runtime layers of agentic processing are provided as shared infrastructure rather than rebuilt by each FIU, with deliberate separation between the two: an industry body operates the governance layer; a separately designated, accredited entity runs the TEE under regulatory oversight.

This appendix addresses the concerns this architectural choice genuinely raises, and the structural properties that hold under it.

An industry body operates the Agent Class Registry, the Processing Receipt Registry, and the Conformance Library (the governance layer).

A separately designated entity, operating under regulatory oversight, runs the TEE that holds AA-fetched data during processing. This was chosen because it is the only architecture in which integrity properties (receipt emission, consent enforcement, conformance certification) can be guaranteed by infrastructure rather than depending on each participant's cooperation.

Receipts emitted by participant-controlled infrastructure are no stronger than internal log files. The framework's value rests on emission being independent of the party whose actions are being recorded. Keeping the rule-setter (the industry body) separate from the TEE operator also keeps ecosystem policy authority structurally distinct from runtime control of the data path.

On the Choice of TEE

The framework adopts TEEs because they provide a practical combination of confidential processing, runtime policy enforcement, and verifiable execution. The framework requires a single execution boundary within which consent constraints can be enforced, agent actions can be

governed, and processing receipts can be generated and cryptographically bound to an attested runtime.

TEEs allow these capabilities to be combined within a single operational model while supporting the latency and scalability requirements of data processing in AA. The choice of TEEs is therefore an architectural decision driven by the framework's requirements for governance, accountability, and runtime assurance.

Concerns this Architecture Raises

- **Concentration risk**

A breach of the central TEE has ecosystem-wide consequences. The blast radius is larger than under a distributed model.

- **Governance capture**

An operator of shared infrastructure has structural power over the ecosystem. The relationship between operations and governance needs careful separation, and both need to remain answerable to regulators.

- **Single point of failure**

Outage of the central TEE halts processing of new fetches across the ecosystem until the TEE recovers. The FIP can still encrypt and dispatch (its wire behaviour is unchanged); the TEE just cannot decrypt until it is back. Availability of the TEE becomes a critical operational requirement, not a best-effort one.

- **Data sovereignty and residency**

Shared processing of Indian financial data raises localisation questions that distributed models naturally avoid.

- **Side-channel risk**

Confidential computing protects enclave memory from direct read but is subject to ongoing security research on side-channel attacks (timing, cache, power, speculative execution) that infer information without reading protected memory.

Mitigations

- **Multi-region TEE deployments with attested failover**

Attested failover across geographically separated TEE instances (within India; see the residency sub-section), each with its own enclave-bound keypair and independent attestation roots. On the FIP side, the FIP's client library publishes the list of currently active enclave public keys and switches automatically if one region is unreachable, so dispatch continues. On the FIU side, a regional failover does not strand the FIU: the dataset is re-encrypted to the FIU's key and forwarded by whichever region completes the work, and the data reference the FIU's agents use resolves to the active region, so both the retention copy and agentic access survive the failover. A single regional incident does not halt the ecosystem on either side.

- **Separation between governance and runtime operations**

An industry body operates the governance layer (registry, Processing Receipt Registry, Conformance Library) and sets the rules these encode. A separately designated entity, accredited under regulatory oversight, runs the TEE that holds and processes AA-fetched data. Regulatory bodies sit above both. No party holds both ecosystem policy authority and runtime control of the data path.

- **TEE-agnostic interfaces**

Receipts, attestations, and consent enforcement defined against a standardised interface so the underlying silicon vendor can change without ecosystem disruption. Reduces lock-in to any one vendor's roadmap or vulnerability history.

- **Regulatory oversight and transparency**

The industry body's governance operations and the TEE operator's runtime operations are both answerable to regulatory bodies, subject to incident disclosure, scheduled audits, and public uptime reporting.

- **Data residency commitments**

TEE instances and all supporting infrastructure are located within India and operate under Indian data protection law. The detailed treatment is in the next sub-section.

- **Side-Channel Mitigations**

The framework addresses side-channel risk through a combination of hardware selection, secure software design, operational controls, and continuous maintenance.

Only confidential-computing platforms that remain under active vendor support are included in the approved hardware list, ensuring that newly disclosed vulnerabilities can be addressed through firmware and microcode updates and that unsupported platforms can be retired from use.

The framework's enclave software follows side-channel-aware coding practices, including constant-time cryptographic operations and deterministic handling of sensitive data, reducing exposure to timing, cache, and speculative-execution attacks.

TEE operator accreditation incorporates operational safeguards such as disabling debug interfaces, restricting performance counters, and enforcing controlled scheduling policies to limit opportunities for exploitation by malicious or compromised operators. Accreditation also includes physical and infrastructure requirements, including deployment in secure facilities, tamper-resistant hardware, and audited access controls, providing additional protection against attacks such as power analysis, electromagnetic leakage, and hardware tampering.

As side-channel attacks remain an active area of security research, the framework does not claim absolute immunity. Instead, it commits to continuously updating its approved hardware list, software standards, and operational requirements in line with evolving industry knowledge and emerging threats.

Data Safety and Residency in India

- **Data residency is non-negotiable**

The industry body operates the governance infrastructure (the registry, the Processing Receipt Registry, the conformance Library) within India and under Indian law; the designated TEE operator runs the TEE on Indian soil under the same legal regime. The specific commitments are as follows.

- **Infrastructure location**

The TEE, the Agent Class Registry, the Processing Receipt Registry, and the Conformance Library are hosted in data centres located in India. Multi-region deployments referenced under the mitigations above are multi-region within India (separate Indian regions, separate Indian data centres), not cross-border. No production data leaves Indian territory at any point in the data path.

- **Sub-processors**

Any service providers engaged for the operation of the TEE or supporting infrastructure are

themselves required to operate within India and under Indian data protection obligations. Sub-processor disclosures are part of the standing transparency reporting of the body that engages them, so the ecosystem and the regulator can verify the chain.

- **Keys and attestation**

The enclave-bound keypair that protects the TEE-side processing copy is generated inside the enclave at boot and never leaves enclave memory. The attestation chain that participants verify before encrypting to that key is anchored to silicon attestation, with the attestation verification service operated by the industry body within India. No key material, no attestation evidence, and no measurement data transits outside Indian infrastructure.

- **Data flows**

AA-fetched data follows a single path under the framework: the FIP encrypts and dispatches to the ecosystem TEE (encrypted to the TEE's enclave-bound public key). The TEE decrypts inside the enclave, hosts agentic processing on the cleartext, and re-encrypts a retention copy to the FIU's key for the FIU to retain as per applicable regulatory and legal requirements. The TEE's outputs (the retention copy, the receipts, and any consent-authorised derivatives) flow only to the FIU named in the consent. All flows transit Indian network infrastructure. Receipts land in the Processing Receipt Registry, which the industry body operates in India. No copy, derivative, or receipt flows outside Indian jurisdiction in the framework's normal operation.

- **Governance**

The industry body's governance operations and the designated TEE operator's runtime operations are governed by Indian regulations and laws and the sector-specific obligations that apply to the AA ecosystem. The TEE operator is accredited under regulatory oversight, subject to scheduled audits and incident disclosure to the relevant Indian regulators. Disputes about data handling are adjudicated under Indian jurisdiction.

- **Auditability**

The framework's audit trail is itself the primary instrument the regulator uses to verify these commitments at runtime. Any deviation from the residency commitments above would surface as a structural anomaly in the receipt stream (a receipt with an unexpected geographic identifier, an attestation that does not chain to the published Indian roots, a sub-processor invocation outside the disclosed list), which is detectable as a violation rather than dependent on after-the-fact reporting.

Position

The shared model is recommended because it is the architecture under which the framework's integrity claims hold: receipt emission, consent enforcement, and conformance certification are guaranteed by infrastructure rather than depending on each participant's cooperation. The concerns above are real, and the mitigations address them. The choice of TEE operator is a regulatory decision the framework does not seek to make on the regulator's behalf, beyond the requirement that the operator be independent of the TSPs, accredited, and supervised.

Appendix B: Glossary

Terms used throughout this paper. Existing AA primitives are grouped first; framework introduced terms follow.

AA Primitives

- **Account Aggregator (AA)**

RBI-licensed consent manager that intermediates the flow of financial information from FIPs to FIUs under signed consent. Does not see or store FI data.

- **Financial Information Provider (FIP)**

Regulated entity holding financial data (bank, NBFC, asset manager, insurer, GST network, depository) that responds to AA-mediated fetch requests.

- **Financial Information User (FIU)**

Regulated entity that consumes AA-fetched financial data to provide services to the Customer (such as lending, advisory, and account aggregation views).

- **Customer**

The individual or entity whose financial information is being shared. Grants consent through the AA, which can be revoked at any time

- **Consent artefact**

Signed AA-issued artefact recording what data may be fetched from which FIPs, for what purpose, how often, and for how long. In this framework, it also carries the PAC block. It is the signed object the TEE verifies and enforces against. Identified by a consent handle.

- **Consent handle**

The opaque identifier for a consent artefact, returned by the AA during the consent flow. It is a reference token, not the signed object: receipts in the Processing Receipt Registry are keyed by it, and the AA queries the Processing Receipt Registry through it. The handle-to-Customer mapping lives only at the issuing AA.

- **Data fetch identifier**

Identifier the AA assigns to each individual fetch under a consent, distinct from the consent handle. The framework uses it to reconcile the delivery and read receipts that bracket a single data placement and first access.

- **Data reference**

Cryptographically bound pointer to a dataset that an FIP encrypted to the ecosystem TEE and the TEE has decrypted inside the enclave. Valid only for the FIU named in the consent and only for the lifetime of the consent. The AA passes the reference to the FIU as part of the consent flow; the FIU's agents access the data through the reference for agentic processing, without the cleartext bytes inside the TEE ever leaving it. The FIU's own retention copy (re-encrypted by the TEE and forwarded as per applicable regulatory and legal requirements) is separate from this reference and is not accessed through it.

- **Central Registry**

Industry-body-operated registry of accredited AA participants (FIPs, AAs, FIUs), including each participant's public key. The TEE fetches the FIU's public key from here to re-encrypt the retention copy. The framework adds the Agent Class Registry alongside it.

AA Primitives

- **Processing Aware Consent (PAC)**

Extension to the AA consent artefact carrying a block that declares what the FIU is permitted to do with the data: which agent classes may act, processing classes from a controlled vocabulary, derivatives that may be retained, transformations explicitly out of scope.

- **Agent class**

A registered software identity for an agent at class granularity (not per instance). An FIU may run many instances of a registered class, but the class is the unit of identification, certification, and accountability. Examples span the risk spectrum: derive-only classes such as Income Assessment, Debt Serviceability, Statement Summariser, and Identity Match Verifier;

decision-support classes such as Liability Estimation and Credit Assessment; orchestration classes such as Underwriting Orchestrator; and consequential autonomous classes such as Decision Synthesiser, Financial Discipline Analyser, Fraud Detector and Disbursement Authoriser.

- **Agent Class Registry (ACR)**

Industry-body-operated registry holding signed manifests for every agent class authorised to operate on AA data, plus MCP server registrations and Conformance certifications. Resolves identity at every API call.

- **Manifest**

Signed declaration registered in the ACR for each agent class: identity, served principals, capabilities, tool surface, model bindings, permitted children, and current conformance status. Uniqueness enforced by hash.

- **Side effect classification**

Three-tier classification of what an agent action does: derive-only (the agent produces derivatives such as scores or summaries but changes no state), internal action (the agent changes state inside the FIU, such as writing to its CRM or queueing a decision), or consequential autonomous decision (the agent commits to a decision with regulatory or customer-facing impact). Derive-only does not mean low impact; a credit score is derive-only. All classes run inside the TEE and emit receipts identically; the classification determines the Conformance rigour the class must pass before production and what the PAC block must explicitly authorise, not where the agent runs or how its receipts are emitted.

- **Processing receipt**

Umbrella term for the four signed receipts the framework introduces: delivery, read, system, and refusal.

- **Delivery receipt**

The receipt emitted by the TEE on receipt of the FIP's ciphertext is called the delivery receipt. The typical contents could include data fetch identifier, payload hash (SHA-256 of the decrypted bytes), FIP identifier, slice descriptor (e.g. account types and date ranges sourced from the book), the data reference under which the dataset is registered in the TEE for agentic access, the FIU re-encryption attestation, decryption timestamp, TEE attestation, FIP signature.

- **Read receipt**

The receipt emitted by the TEE when the FIU first accesses the placed data through its references is called the read receipt. The typical contents could include data fetch identifier (matches the delivery receipt's), consent handle, payload hash (verified equal to delivery hash), accessing FIU identifier, first-access timestamp, TEE attestation, reference to the delivery receipt.

- **System receipt**

The receipt which maintains a record of what was read, by which agent class, the tool calls by argument and result hash, and the output classification is the system receipt. The typical contents could include consent handle, agent class identifier and manifest version, operator identifier, served principal (the FIU) identifier, parent receipt reference (for child agents), source artefact hashes referencing the read receipt, processing class from the PAC vocabulary, side effect classification, tool calls (MCP server identifier, tool name, argument hash, result hash), output classification, access count for the consent so far, Conformance certification status at time of action, TEE attestation, action timestamp.

- **Refusal receipt**

The refusal receipt is a receipt emitted by the TEE when it blocks an action before execution because the agent class or the processing it attempted is not permitted by the consent's PAC block, or the consent is revoked or expired. It ensures that any action that was attempted and prevented is logged in the receipt tree as the action so as to maintain transparency. The typical contents could include consent handle, agent class identifier and manifest version, processing class attempted, reason code, parent receipt reference, TEE attestation, timestamp. Enforcement happens at the TEE before the action; the Processing Receipt Registry verifies after.

- **Lifecycle receipts (termination, expiry, deletion)**

Receipts that close out a consent rather than record a live action. A termination receipt is emitted when an in-flight invocation is wound down because the consent was revoked mid workflow; an expiry receipt when the consent reaches the end of its declared data life; a deletion receipt as proof when a dataset or derivative is disposed of under the retention rules.

- **Customer receipt**

Rights artefact derived from system receipts, issued to the Customer summarising processing in plain language: which agents acted, what categories of data they read, what derivatives were produced. The typical contents could include consent handle and grant date, FIU identity in plain language, list of agent activities described non-technically (e.g. "income pattern analysis", "affordability assessment", "final lending decision"), categories of data accessed (e.g. "24

months of transactions across three bank accounts"), derivatives produced and retained, total access count, retention status and expected deletion date, link to file a grievance.

- **Processing Receipt Registry**

Industry-body-operated Registry of all receipts across the ecosystem, used for audit, grievance, regulator queries, and structural detection of coverage gaps. The live runtime Registry of the framework, itself hosted inside a TEE so its operator cannot read or alter the Registry at runtime. Holds no personal data: receipts are keyed by the opaque consent handle, whose mapping to an identified Customer lives only at the issuing AA, and receipt contents are hashes, references, and classification labels rather than raw data or direct identifiers. Queried by consent handle, never by consumer identity.

- **Conformance Library**

Industry-body-curated, versioned library of attack patterns and test cases that every agent class must pass before being authorised for consequential decisions. The certification test bench, distinct from the Processing Receipt Registry (which is the runtime Registry). Test categories include schema and format conformance (manifest and receipt validation), behavioural conformance (revocation enforcement, scope adherence, retention behaviour at consent expiry, cross-customer leakage in shared TSP stacks), conformance (resistance to prompt injection in user inputs and tool outputs, resistance to tool-chain abuse, parent refusal of children that escalate scope), and operational conformance (key management, sub-processor disclosure, incident response). New attack patterns are sourced from SAFE MCP, OWASP LLM Top 10, MITRE ATLAS, and ecosystem incident learnings.

- **Technology Service Provider (TSP)**

Vendor that operates an agent stack on an FIU's behalf (Pattern B). TSP-FIU binding is expressed via served principals in the TSP's manifest plus counter signature from each FIU in the ACR.

- **Operator patterns**

Three deployment patterns differ in registration shape: who signs the agent class manifest as operator, who counter-signs as served principal, and whether the model provider is a signing party in that registration or only a referenced one. In all three, the agent runs inside the TEE, the model runs outside it, and the TEE emits and signs every receipt.

Pattern A: the FIU is both operator and served principal; the registration has one signing party.

Pattern B: a TSP signs as operator and the FIU counter-signs as served principal; receipts reference both distinctly.

Pattern C: the FIU signs as both operator and served principal (as in A), but the model is on an accredited AI model infrastructure; the provider is referenced in the receipt as the destination model, as a signing party in the registration.

- **Served principals**

Field in a TSP agent's manifest listing the FIU identifiers it is authorised to act on behalf of. Each FIU named must counter sign the binding in the ACR before it is active.

- **MCP wire observer**

Process the TEE attaches to every agent, deriving receipt entries from observation of the MCP protocol wire between the agent and its tools, by a process the agent cannot signal to or suppress. Because all agentic processing runs inside the TEE, it is the single emission mechanism for all receipts.

- **Sandbox**

Industry-body-operated environment for experimentation: synthetic or anonymised data only, a separate registration class with Conformance certification deferred, and explicit time and volume caps. The on ramp for new agent classes before production.

References

This section lists the foundational documents that ground the framework's technical, regulatory, and adversarial-testing claims. References are organised by topic rather than cited inline.

Regulatory Framework

ReBIT. (2023, August 9). Account Aggregator Ecosystem API Specifications. https://api.rebit.org.in/

Sahamati. (2026, April 6). Home - Sahamati. https://sahamati.org.in/

Confidential Computing

Michalevsky, Y. (2026, February 23). Building a Dark pool on Stellar: MPC, FHE, and TEES compared. Stellar. https://stellar.org/blog/developers/building-a-dark-pool-on-stellar-mpc-fhe-and-tees-compared

NVIDIA Trusted Computing Solutions - NVIDIA Docs. (n.d.). NVIDIA Docs. https://docs.nvidia.com/nvtrust/index.html

Confidential AI

Kapilv. (n.d.). Confidential AI - Azure Confidential Computing. Microsoft Learn. https://learn.microsoft.com/en-us/azure/confidential-computing/confidential-ai

Adversarial AI Testing

Kautz, F., Subedi, A., Bista, B., & Adusumilli, V. (Ravi). (2025, January 21). Security Analysis Framework for Evaluation of MCP and AI Agents. SAFE-MCP. https://www.safemcp.org/

MITRE ATLAS. (n.d.). ATLAS Matrix for AI Systems. https://atlas.mitre.org/

OWASP. (n.d.). OWASP Top 10 for Large Language model Applications. https://owasp.org/www-project-top-10-for-large-language-model-applications/

Authentication

Cappalli, T., Kumar, A., Lundberg, E., Miller, M., Pascoe, & Satragno, N. (Eds.). (2026, May 26). Web Authentication: An API for accessing Public Key Credentials Level 3. World Wide Web Consortium; World Wide Web Consortium. https://www.w3.org/TR/webauthn-3/

Shamas, M. (2026, April 29). FIDO User Authentication Specifications Overview. FIDO Alliance. https://fidoalliance.org/specifications/

Model Context Protocol

MCP. (n.d.). What is the Model Context Protocol (MCP)? Model Context Protocol. https://modelcontextprotocol.io/docs/getting-started/intro

Side Channel Mitigations

Google. (n.d.). Protecting the world's data with industry-leading security. Google Data Centers. https://datacenters.google/advancing-security/

Google Cloud. (n.d.). How Google protects the physical-to-logical space in a data center. Google Cloud Documentation. https://docs.cloud.google.com/docs/security/physical-to-logical-space