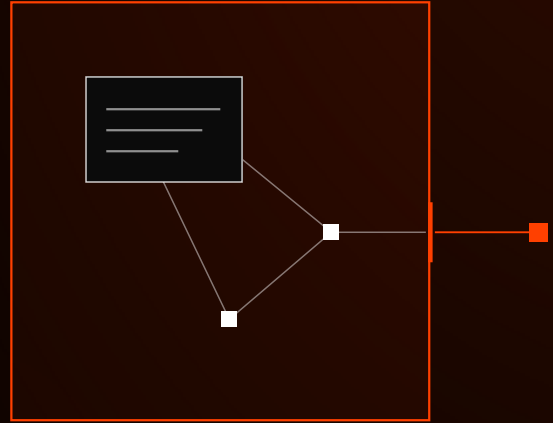




How Can Financial Institutions Put AI to Work With Clear Controls?

A Practical Checklist for AI Access and Actions in Digital-Asset Operations

AI assistants can help teams find information, investigate discrepancies and troubleshoot systems. In digital-asset operations, that work can reach client records, production systems and wallet-provider services.

The task may be narrow while the credentials permit much more. An instruction to investigate does not restrict a credential that also permits transfers.

Use this checklist to identify what can proceed under existing controls, what needs adjustment and which access should remain limited until verified.

Who this is for

For security, technology and operations teams at exchanges, custodians, trading firms, asset managers, payments providers and banks working with digital assets.

One task does not grant every permission

Illustrative example: an assistant investigates a ledger discrepancy.

Within the task

- Read approved records for specified accounts
- Compare with the ledger
- Prepare an internal summary for the approved audience

Requires separate authority

- Initiate a transfer
- Change account settings
- Close the exception
- Contact a client

The same distinction applies when an assistant reads workplace documents or inspects code and logs: sharing externally or changing production requires separate authority.

Start with one existing or proposed workflow

Define the workflow and ownership

Identify its tools, credentials and systems. Name who owns the workflow, maintains permissions and connections, approves changes and removes access when people change roles or agents are retired.

Use the checklist with the relevant teams

Use the questions with the relevant teams, alongside existing privacy, supplier and model-risk reviews. One reviewed workflow does not cover all AI use.

[Jump to a section](#)
[Checklist](#)

03

[Tests](#)

05

[What to do next](#)

06

[References](#)

06

1 Theme 1 of 6

Give AI the access its task requires

01 Can you tell who is acting?

Identify the employee or service initiating work and whether the connection acts as that person, a service or a delegated bot. Account for additional agents and shared credentials.

02 Does access match the task?

Limit accounts, records and operations. Scope reads and writes separately for each environment. Read-only access still needs limits on client records, positions and transaction history. Check actual permissions. [1]

2 Theme 2 of 6

Control credential use as well as storage

03 Are credentials protected from exposure?

Keep raw credentials out of model context and unauthorised tool responses. Check files and environment variables, and limit who can retrieve secrets.

Also distinguish API authentication credentials from asset-signing keys. A protected secret can still authorise an action outside the task.

04 Can credentials expire and access be revoked?

Use scoped, short-lived access where supported; manage longer-lived credentials through their lifecycle. Test revocation at the connected service, including cached tokens and active sessions. Record any remaining access or delay.

3 Theme 3 of 6

Preserve authority at the point of action

05 Are permissions enforced outside the model?

Check each requested operation before execution. Use relevant team, role, resource, environment, time and transaction conditions. Build on native service controls and verify what they cover. [2]

06 Can you prove that a denied request did not execute?

Test an unapproved operation and an out-of-scope resource. Check the target system, rather than relying only on a denial message or policy setting.

Wallet-provider example

Check the provider's own permissions and transaction approvals. Fireblocks documents an MCP server with writes disabled by default and recommends minimum API permissions and human oversight through its Policy Engine. This is an implementation example, not a description of every customer's setup. [4]

4

Theme 4 of 6

Keep approvals meaningful in daily operations

Decide which actions may run within policy and which require human approval.

07

Does approval cover the actual business action?

Show the outcome and key details: for example, source, destination, asset and amount, or a production target and proposed change. Bind approval to the actor and exact action. Check expiry and reuse; changed significant parameters require fresh approval. [3]

08

Can approvals work without weakening existing controls?

Define authorised approvers and backups. Preserve required separation of duties and four-eyes approval; the agent must not approve itself. Test delays and failure handling so routine work does not depend on bypassing controls.

5

Theme 5 of 6

Check the complete access path

09

Are permitted data and destinations clear?

Limit which client records and operational data may enter model context, reports, messages and logs. Reading information does not grant authority to email it externally or change sharing permissions.

10

Can another route bypass the controls?

Check connectors, direct APIs, local credentials and terminal access. Identify coverage for prompts, responses and tool calls. Follow existing app-approval rules; restrict administrative rights and log policy changes.

6

Theme 6 of 6

Make actions attributable and access stoppable

11

Can you reconstruct what happened?

Link the actor, agent, operation, target, time, policy decision, approval and execution result, including denials and failures. Approval alone does not prove success.

For ongoing review, agree how long to retain logs, keep them retrievable and protect them from alteration. Exclude or redact secrets and sensitive client information.

12

Can an accountable operator stop further access?

Test new requests and active sessions. Name the owner for intervention, escalation and recovery. Stopping future calls does not undo an action already completed.

Test before expanding access

Use test data and a sandbox first. Confirm permitted work succeeds, then test the restrictions below. Before production, verify the intended identities, permissions and policies across the actual access paths.

Test	Expected result
Out-of-scope operation or resource	Denied; the target does not execute it.
Declined, expired or changed approval	No execution without valid approval for the exact action.
Revocation or operator intervention	Further requests and active sessions behave as policy requires.
Control failure or alternative route	No silent access; emergency routes are separately authorised and logged.
Misleading instructions in retrieved content	The attempt cannot exceed the authorised action or data boundary.

Repeat relevant tests when connections, permissions, approval rules, credentials or AI components materially change. Tests establish evidence for the scenarios tried, not every failure. Observation mode does not itself block actions; set permissions before connecting a pilot. [3]

What to do next



Controls meet the task's needs.

Use the evidence in your normal review to decide whether to proceed within the verified scope.



Controls need adjustment.

Narrow permissions, clarify approvals or improve records.
Assign owners and retest before expanding affected access.



An important restriction is unverified or a required control fails.

Keep affected access disabled or limited until resolved. Permitted work must stay within verified controls and normal approvals.

For any service handling credentials or authorisation, check secret storage, administrative access, independent assurance and recovery arrangements through normal supplier review. Agree the permitted scope and ownership of remaining actions.

Selected references

- 1 [NIST SP 800-53 Revision 5](#)
- 3 [OWASP AI Agent Security](#)

- 2 [OWASP Excessive Agency](#)
- 4 [Fireblocks MCP server documentation](#)