



Developer data-path pilot pack

CISO and engineering leader

[Read the complete guide on aona.ai](#)

Start by mapping how a developer's task sends data to AI: typed prompts, selected code, repository context, terminal output and connected tools. Approve the data and account, choose controls for each path, then run a small synthetic pilot. A tool name or no-training statement alone does not establish that company data stays within your intended boundary.

Select one task, assign its control owners and record a scoped pilot decision.

Synthetic planning examples. No installed-product tests or results are included.

Your review

Reviewer: _____ Date: _____

Organisation or team: _____

☐ Define the task and environment

Record the approved account, client/version, OS, execution location and repository owner.

☐ Give every input path an owner

Separate client access, provider terms and supported endpoint policy evaluation.

☐ Keep planned and observed results separate

Use untested until evidence exists; do not treat a marker as a realistic secret-detector test.

Checked items record your review, not a compliance approval or an installed-product test result.

Notes and next actions

Decision and scope: _____

Evidence to collect: _____

Owner and next review date: _____

Review matrix

Input source	Client/runtime	Recipient and retained copy to review
Synthetic code	Selected editor or CLI	Model destination and transcript
Fake secret marker	Supported test input path	Control boundary before submission
Synthetic test trace	Paste or file upload	Provider request and retained history
Repository marker	Context or file-reading tool	Tool result and downstream context

Developer data-path pilot

All values are synthetic. Use an isolated test repository and an approved test account. No file runs commands, contacts a service or configures a product.

1. Choose one task in the “Pilot decision” section.
2. Complete the client and owner fields in data-path-register.csv.
3. Select only the relevant labels from synthetic-pilot-markers.json. They are not real credentials and may not trigger a data classifier. Agree a supported test rule or representative safe fixture with its owner.
4. If you manually test an approved AI client, use only these synthetic inputs. Record what actually happened and preserve the exact scope.
5. Route deeper client-specific questions to the linked guides; do not broaden the pilot by adding new accounts or connectors.

Data path register

These records are arranged vertically for reading and printing. The Excel workbook retains the full column layout.

Record 1

Field	Record value
Task	Explain a synthetic test failure
Input source	toy test trace
Input path	pasted log
Data owner	ASSIGN
Client and version	RECORD
Execution location	RECORD
Recipient or destination	RECORD
Retained copy to review	client and provider history
Control boundary	DEFINE
Intended control	SELECT
Expected outcome	DEFINE
Observed outcome	UNTESTED
Evidence	

Record 2

Field	Record value
Task	Explain a synthetic function
Input source	toy function
Input path	selected code
Data owner	ASSIGN
Client and version	RECORD
Execution location	RECORD
Recipient or destination	RECORD
Retained copy to review	client and provider history
Control boundary	DEFINE
Intended control	SELECT
Expected outcome	DEFINE
Observed outcome	UNTESTED
Evidence	

Record 3

Field	Record value
Task	Find a synthetic marker
Input source	fixture folder
Input path	repository context
Data owner	ASSIGN
Client and version	RECORD
Execution location	RECORD
Recipient or destination	RECORD
Retained copy to review	tool result and downstream context
Control boundary	DEFINE
Intended control	SELECT

Field	Record value
Expected outcome	DEFINE
Observed outcome	UNTESTED
Evidence	

Record 4

Field	Record value
Task	Review a connected response
Input source	approved synthetic service data
Input path	approved tool only
Data owner	ASSIGN
Client and version	RECORD
Execution location	RECORD
Recipient or destination	RECORD
Retained copy to review	service log and model transcript
Control boundary	DEFINE
Intended control	SELECT
Expected outcome	DEFINE
Observed outcome	UNTESTED
Evidence	

Synthetic pilot markers

Notice: Synthetic labels only. No credential, customer record or test result.

Code: `def synthetic_total(values): return sum(values)`

Secret: NOT_A_CREDENTIAL_D01_CANARY

Log: SYNTHETIC_D01_TEST_TRACE

File test_total.py, line 6: `synthetic_total(None)`

File example.py, line 2: `return sum(values)`

TypeError: NoneType object is not iterable

Expected test behaviour: return 0 for an empty input; decide separately whether None is valid.

Context: SYNTHETIC_D01_CONTEXT_MARKER

Pilot decision

Task: _____
Business purpose: _____
Permitted inputs: _____
Restricted inputs: _____
Account, client/version and OS: _____
Local, remote or cloud execution: _____
Expected control and prerequisite: _____
Observed result: UNTESTED
Evidence location: _____
Decision: NOT YET REVIEWED
Owner and review date: _____
Recheck triggers: new client, model, connector, repository location or policy.

Sources and scope

Sources checked 2026-09-21. Review the current requirements and your actual agreement before using this workbook for a real decision. Native controls vary by provider, release and input path. Agent inspection is limited rollout; this pack does not imply agentless cloud protection.

[Cursor: Privacy and Data Governance](#). Distinguishes local-client data requests, provider handling and cloud-agent environments.

[OpenAI: Agent approvals and security](#). Separates sandbox, approval and network controls for local and cloud use.

[Aona: AI security coverage](#). Current public distinction between supported endpoint policy paths and limited-rollout agent inspection.