NLT Communication & Control Flows

PE evaluate · POC ship · memory bridge · quality gate

See the machine move.

Four sequences every operator traces on day one.

AUDIENCE Architects · operators

Executive summary

4 CORE FLOWS PE · POC · memory · quality	1 EXECUTOR AgentForge task pipeline	HTTP MEMORY Bridge-only contract	Vol 3 PREREQUISITE Integration map

This volume documents the four control flows architects and operators trace most often when onboarding to the NLT platform. Vol 3 covers static integration boundaries; this volume covers motion.

Key findings

1 PE evaluate — intake through FUND verdict.	2 POC ship — FUND through builder to portfolio.nltlabs.ai.
3 Memory bridge — session recall via NLT Quality Pipeline HTTP.	4 Quality gate — dev-time checker loop on every PR.

Motion vs map
Read Vol 3 first for trust boundaries. Use this volume when onboarding operators who need to trace live workflows.

SECTION I

Flow overview

The four sequences and how they connect.

- PE evaluate — idea intake through FUND verdict.
- POC ship — FUND through builder to portfolio.nltlabs.a
- Memory bridge — session recall and reinforce via NLT
- Quality gate — tapps-mcp dev-time checker loop on every PR.

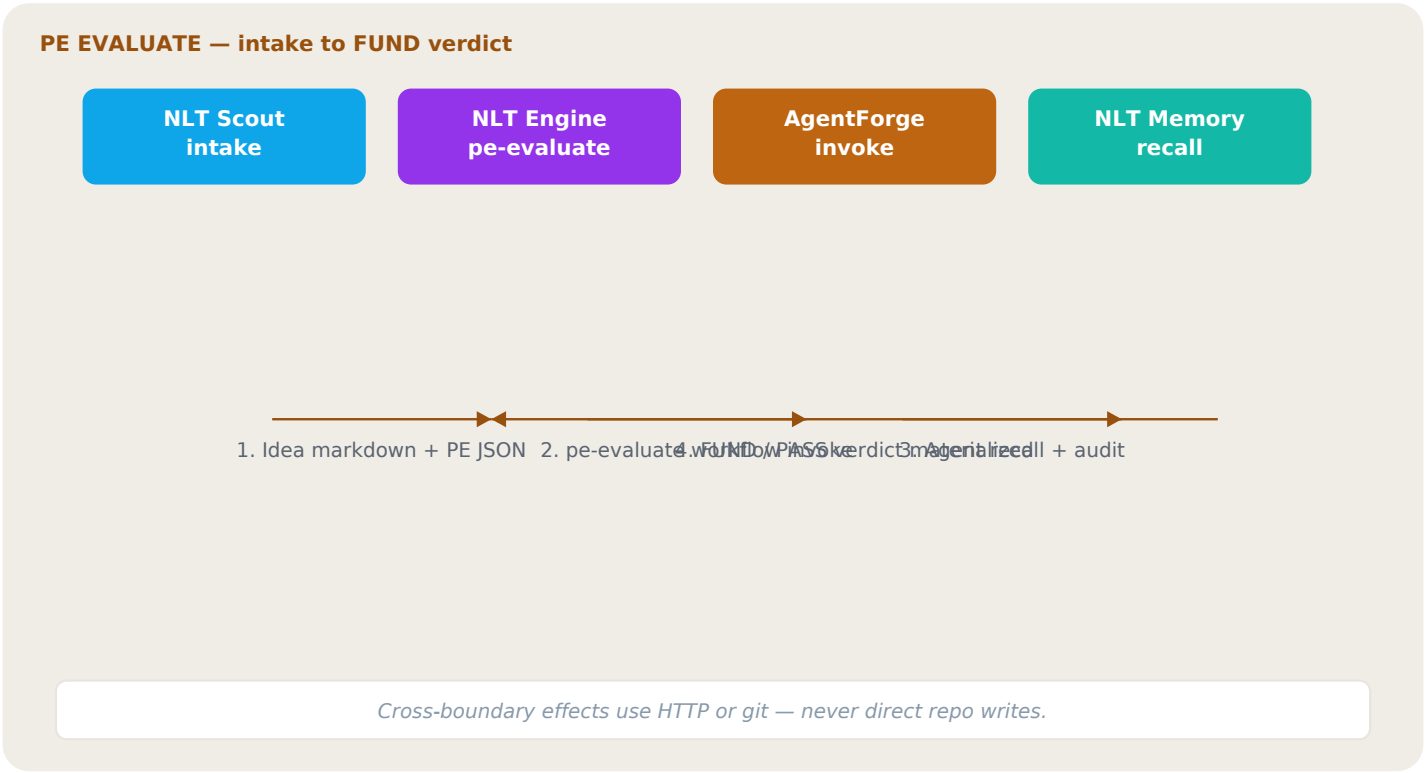
Naming
Diagrams use outward component names from docs/NLT_BRAND.md.

SECTION II

PE evaluate sequence

NLT Scout promotes intake artifacts. NLT Engine runs pe-evaluate via AgentForge with optional NLT Memory recall before materializing a FUND or PASS verdict.

Operators trace this flow when onboarding a new idea slug: confirm intake files exist under NLT Scout, verify the pe-evaluate workflow is registered in NLT Engine, and watch for a FUND row in the portfolio registry before scheduling builder capacity.



Sequence — PE evaluate sequence

Step	Actor	Action	Artifact
1	NLT Scout	Capture intake markdown and PE JSON	ideas//
2	NLT Engine	Queue pe-evaluate workflow	workflows/pe-evaluate.yaml
3	AgentForge	Invoke executor with policy + SBOM	task audit log
4	NLT Memory	Recall prior PE patterns (optional)	tapps_memory search
5	NLT Engine	Materialize FUND or PASS verdict	portfolio registry row

Operator notes

- FUND unlocks poc-ship; PASS defers builder spend.
- Memory recall is optional but improves PE consistency across slugs.
- AgentForge audit logs are the source of truth for invoke failures.
- Cross-repo writes stop at each lane boundary — use git for handoffs.

pe-evaluate — workflow depth

The PE pipeline fans out to independent research analysts, then aggregates into a FUND / PASS / DIG verdict. Production manifest uses on_error: partial so one analyst failure does not abort the entire evaluation.

pe-evaluate — node map

Node	Kind	Agent
market	agent	nlt-pe-market-researcher
comp	agent	nlt-pe-competitive-analyst
feas	agent	nlt-pe-feasibility-analyst
fin	agent	nlt-pe-financial-modeler
reg	agent	nlt-pe-regulatory-analyst
creative	agent	nlt-pe-creative-director
dev	agent	nlt-pe-devils-advocate
auditor	agent	nlt-number-auditor
firm	agent	nlt-pe-firm

Error paths

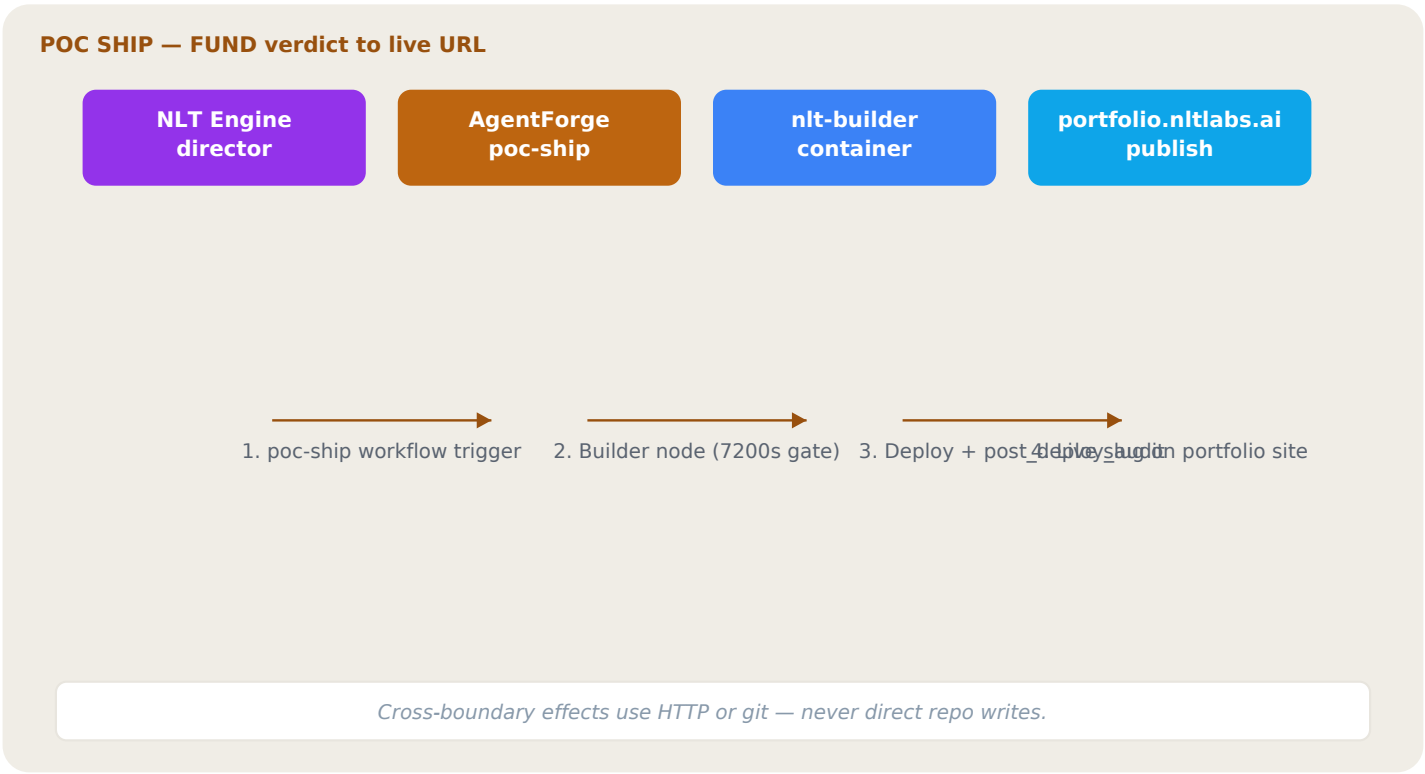
Failure	Behaviour	Operator action
Analyst timeout	on_error: partial drops node data	Re-run pe-evaluate or single analyst
Insufficient citations	assessment_status: insufficient_input	Feed richer intake markdown
Invoke 403	Bearer / policy mismatch	Verify AgentForge project registration
Brain recall empty	Optional — PE proceeds without memory	tapps_memory search + reinforce

SECTION III

POC ship sequence

After a FUND verdict, poc-ship orchestrates builder nodes, deploy audit, and quality review until the slug is live on portfolio.nltlabs.ai.

This is the capital-intensive path: only FUND slugs enter poc-ship. The builder node enforces a long-running gate; post_deploy_audit must pass before the public URL is advertised to investors.



Sequence — POC ship sequence

Step	Actor	Action	Artifact
1	NLT Engine	Director accepts FUND slug	registry FUND row
2	AgentForge	Run poc-ship workflow nodes	builder + deploy audit
3	nlt-builder	Build container image (7200s gate)	OCI artifact
4	NLT Engine	Publish slug to portfolio site	portfolio.nltlabs.ai/
5	Operator	Verify live URL + post_deploy_audit	public proof link

Operator notes

- Builder substrate failures roll back to NLT Engine director queue.
- Deploy audit captures HTTP smoke + security posture for the slug.
- Live URL is the investor-facing artifact — not an internal staging host.
- Quality review may invoke tapps-mcp gates on the shipped repo.

poc-ship — workflow depth

After FUND, poc-ship scopes builder work, runs convergence loops between design/QA pairs, and gates deploy on post_deploy_audit before portfolio URL advertisement.

poc-ship — node map

Node	Kind	Agent
director	agent	nlt-poc-director
market_pricing	agent	nlt-market-pricing-researcher
product_designer	agent	nlt-poc-product-designer
visual_assets	agent	nlt-poc-visual-assets
designer	agent	nlt-poc-designer
copywriter	agent	nlt-poc-copywriter
gtm_operator	agent	nlt-gtm-operator
gtm_qa	agent	nlt-poc-gtm-qa
visual_qa	agent	nlt-poc-visual-qa
content_qa	agent	nlt-poc-content-qa
builder	agent	nlt-poc-builder-gateway
post_deploy_audit	agent	nlt-poc-post-deploy-audit
quality_reviewer	agent	nlt-poc-quality-reviewer

Loop	Members	Max iter	Convergence
visual_regen	visual_assets, visual_qa	3	visual_qa.converge
content_regen	copywriter, content_qa	3	content_qa.converge
gtm_regen	gtm_operator, gtm_qa	3	gtm_qa.converge

Error paths

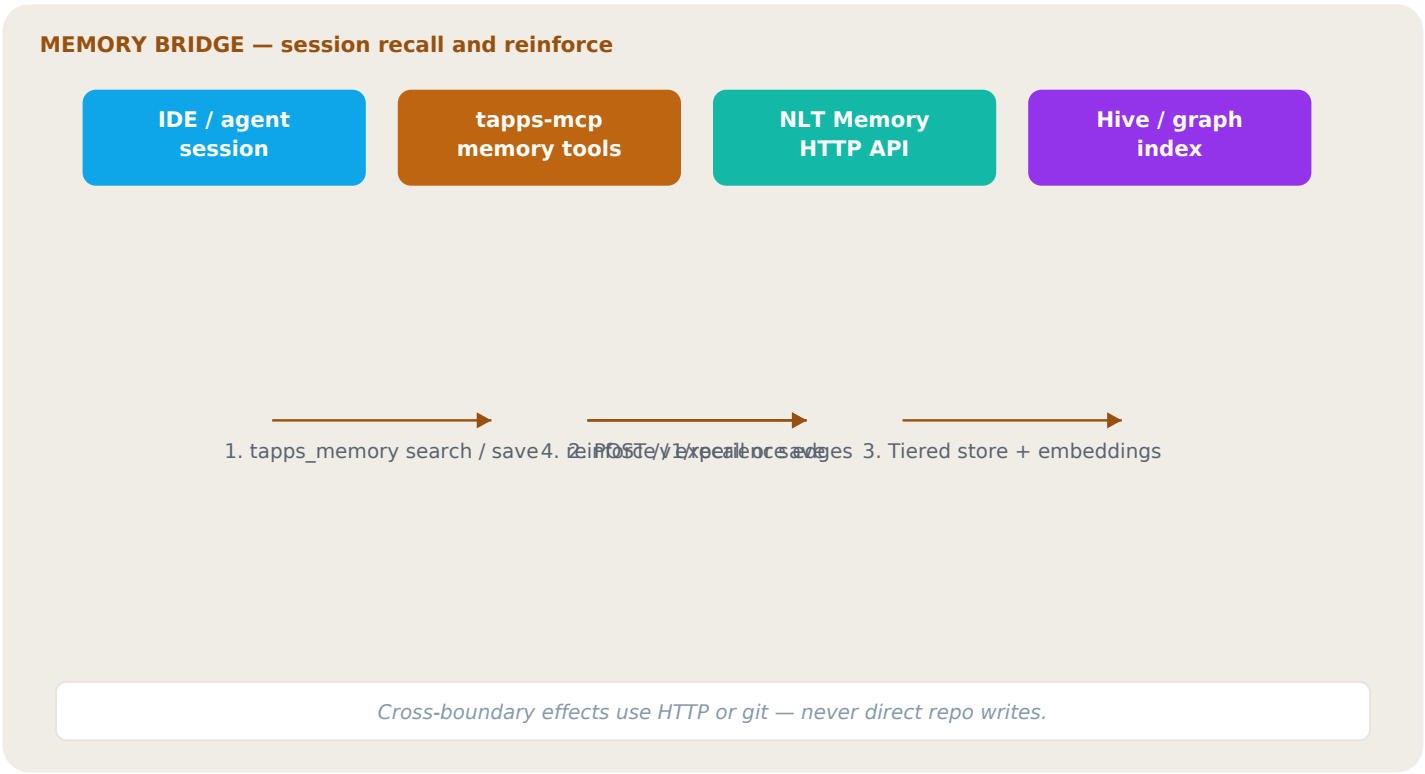
Failure	Behaviour	Operator action
Builder timeout (7200s)	Workflow partial or abort per on_error	Inspect nlt-builder logs
Deploy audit fail	Slug withheld from portfolio	Fix HTTP smoke / security posture
Loop max iterations	Judge does not converge	Director narrows scope or aborts ship
Non-FUND decision	Director refuses downstream nodes	Return to pe-evaluate

SECTION IV

Memory bridge sequence

Consumer repos never embed NLT Memory in-process. NLT Quality Pipeline memory tools negotiate HTTP for recall, save, reinforce, and hive propagation.

Every agent surface — Cursor, Claude Code, CI bots — uses the same bridge. Session recall improves repeatability; reinforce and experience edges feed the graph index for cross-agent handoff.



Sequence — Memory bridge sequence

Step	Actor	Action	Artifact
1	IDE session	Agent calls tapps_memory	MCP tool invocation
2	NLT Quality Pipeline	Negotiate HTTP bridge to brain	TAPPS_BRAIN_AUTH_TOKEN
3	NLT Memory	Search / save / reinforce tier	Postgres + embeddings
4	NLT Memory	Update graph / hive edges	experience + reinforce
5	Consumer repo	Never embed brain in-process	HTTP-only contract

Operator notes

- Set TAPPS_BRAIN_AUTH_TOKEN in MCP host environments.
- Prefer memory search before re-deriving architecture decisions.
- Architectural tier entries may supersede prior versions automatically.
- Hive propagation is opt-in for multi-agent teams.

Memory bridge — operator depth

NLT Memory is never embedded in consumer processes. NLT Quality Pipeline negotiates HTTP for search, save, reinforce, hive propagation, and knowledge-graph queries. The bridge health block in tapps_session_start mirrors brain-native diagnostics.

Memory tool clusters

Tool cluster	Purpose	When operators use it
tapps_memory search/recall	Ranked BM25 + composite scoring	Before re-deriving architecture
tapps_memory save	Tiered retention (architectural/pattern)	After sign-off decisions
tapps_memory reinforce	Boost confidence on proven entries	After successful ship
hive_propagate	Cross-agent memory fan-out	Multi-agent teams only

Error paths

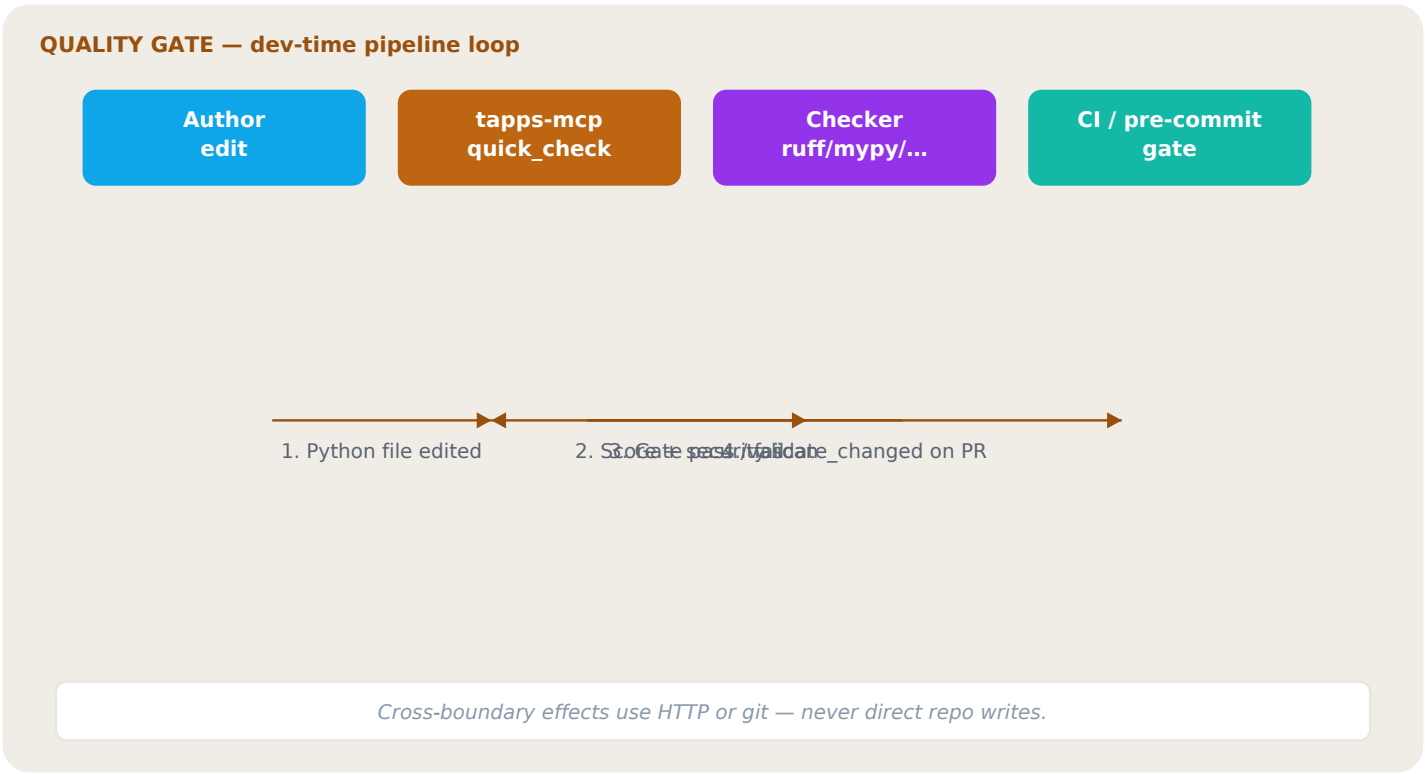
Failure	Behaviour	Operator action
brain_auth_failed	Bridge degrades; memory tools error	Set TAPPS_BRAIN_AUTH_TOKEN; tapps doctor
Empty search	No ranked hits above threshold	Save architectural entry; reinforce after use
Contradiction detected	Supersede or reconcile tier	tapps_memory contradictions + gc
Pool / circuit open	HTTP bridge backs off	Check NLT Memory health + queue depth

SECTION V

Quality-gate dev flow

Authors run tapps_quick_check after edits; CI and pre-commit hooks batch validate_changed on pull requests before merge.

The gate is dev-time discipline — it prevents undeclared task completion and catches security regressions before merge. Consumer repos wire hooks via tapps_init / tapps_upgrade scaffolding.



Sequence — Quality-gate dev flow

Step	Actor	Action	Artifact
1	Author	Edit Python in consumer repo	git diff
2	NLT Quality Pipeline	tapps_quick_check per file	score + gate + security
3	Checker stack	ruff, mypy, bandit, radon	preset threshold
4	CI / hook	tapps_validate_changed on PR	blocking merge
5	Operator	tapps_checklist before ship	task-type checklist

Operator notes

- quick_check runs score + gate + security on one file.
- validate_changed batches git-changed Python with explicit file_paths.
- Pre-commit may block commits; CI blocks merge on gate failure.
- TAPPS_SKIP_GATE=1 is for emergencies only — audit when used.

Quality gate — operator depth

Dev-time gates in NLT Quality Pipeline are distinct from runtime NLT Memory. Authors run quick_check after each Python edit; validate_changed batches git diffs before merge. Consumer repos wire hooks via tapps_init / tapps_upgrade scaffolding.

Gate stages

Stage	Tool	Blocks when
Per-file edit	tapps_quick_check	Score below gate or security fail
Pre-merge	tapps_validate_changed	Any changed file fails gate
Pre-commit	NLT Quality Pipeline validate-changed --quick	Staged Python below threshold
Session end	tapps_checklist	Required pipeline tools skipped

Error paths

Failure	Behaviour	Operator action
Lint blocker (E501, F401)	Gate score forced to 0	Fix ruff issues; re-run quick_check
Security CVE in dep	security_passed false	pip-audit remediation or pin bump
Checker missing in CI	Degraded score / doctor warning	NLT Quality Pipeline doctor --quick in CI image
TAPPS_SKIP_GATE=1	Hook bypass — telemetry only	Document emergency; fix root cause

AgentForge task pipeline (detail)

The diagram below is the generic AgentForge executor path — identity, policy, SBOM, executor, and audit — referenced by pe-evaluate and poc-ship nodes. Every workflow invoke traverses the same audit surface even when node labels differ.

Platform purity means consumer repos never patch AgentForge platform trees. Per-project state lives under consumer checkouts; the executor remains agnostic to NLT Scout or NLT Engine folder names.

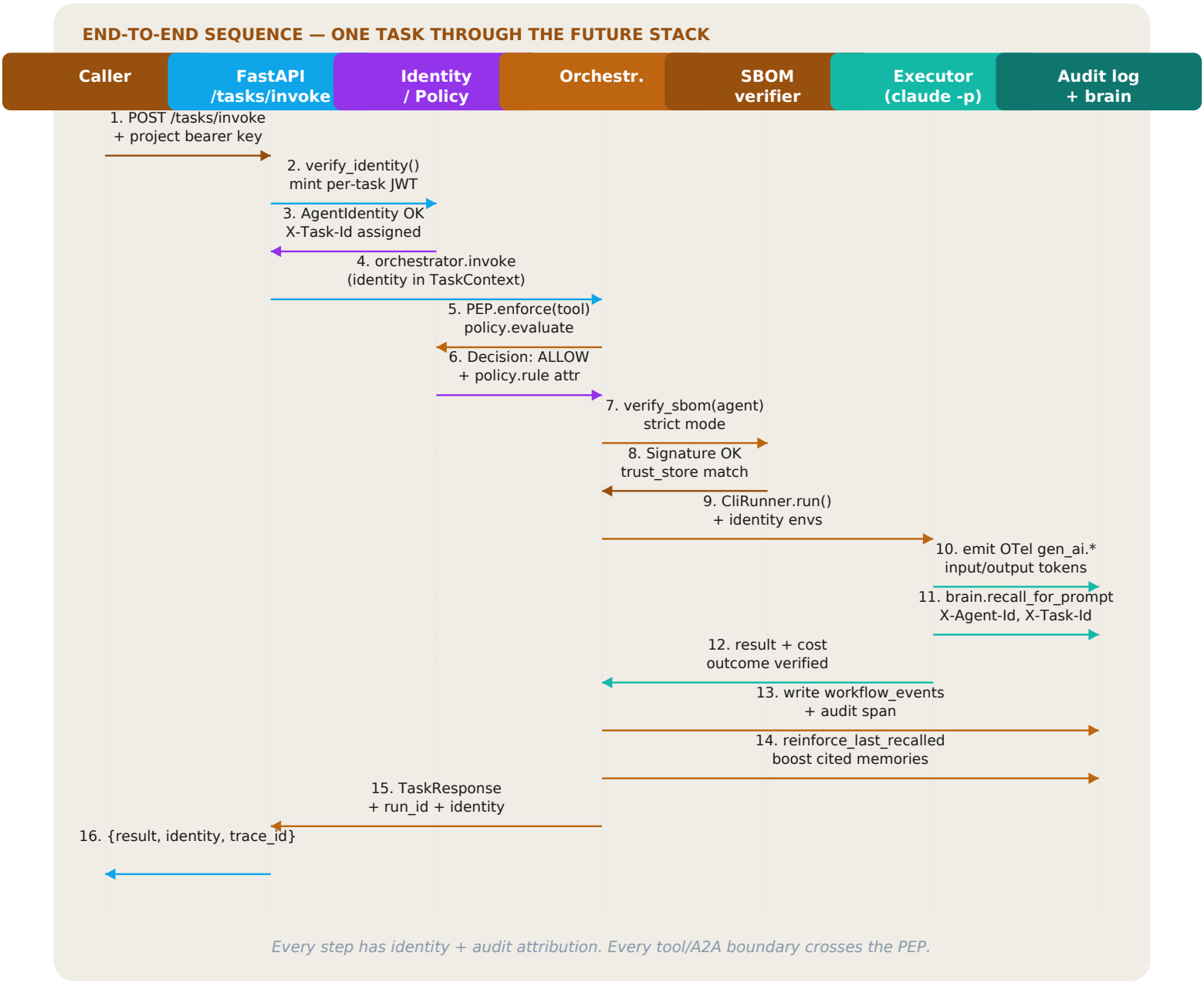


Figure — generic AgentForge executor path

Step	Actor	Action	Artifact
1	Caller	HTTP invoke with project bearer	Authorization header
2	AgentForge	Resolve identity + policy bundle	afp_ scope
3	AgentForge	Attach SBOM + audit context	task record
4	Executor	Spawn Claude CLI or ContainerRunner	subprocess / OCI
5	AgentForge	Persist audit + experience index	NLT Memory bridge

- Identity: project bearer maps to afp_ private memory scope.
- Policy: workflow YAML + agent definitions loaded from NLT Engine.
- Audit: every invoke persists task id for operator forensics.
- See Vol 5 (AgentForge deep dive) for workflow inventory.

SECTION VI

Glossary

Terms used across control-flow diagrams.

Platform glossary

Term	Definition	Related flow
FUND verdict	NLT Engine schema value meaning proceed to POC build.	Flow II
Memory bridge	NLT Quality Pipeline HTTP path into NLT Memory for session recall.	Flow III
PE JSON	Portfolio-entry artifact produced by NLT Scout intake.	Flow I
Platform purity	AgentForge rejects consumer-specific code in platform tree.	AgentForge detail
Quality gate	NLT Quality Pipeline score threshold blocking undeclared task completion.	Flow IV
Trust boundary	Component that owns writes; cross-effects use HTTP or git.	Vol 3
PASS verdict	PE outcome that defers builder spend until criteria change.	Flow I
post_deploy_audit	HTTP + security smoke run before public URL is advertised.	Flow II

SECTION VII

Scout → Engine intake dispatcher

Git handoff before pe-evaluate invoke.

NLT Scout writes durable markdown and PE JSON under ideas/<slug>/. NLT Engine intake_dispatcher polls those artifacts and promotes slugs into the pe-evaluate queue — Scout never writes Engine registry rows directly.

Intake dispatcher sequence

Step	Actor	Artifact	Gate
1	NLT Scout	IdeaCandidate + idea markdown	Vertical workflow complete
2	Git push	Consumer repo	Review + branch policy
3	intake_dispatcher	PE JSON path	Schema + slug uniqueness
4	NLT Engine	pe-evaluate invoke	AgentForge registration OK
5	AgentForge	Audit task id	Policy + bearer scope

Error paths

Failure	Behaviour	Operator action
Missing PE JSON	Dispatcher skips slug	Re-run Scout export workflow
Schema validation fail	Artifact quarantined in logs	Fix JSON against Engine contracts
Duplicate slug	No second pe-evaluate row	Merge or rename idea folder
Invoke 403	No AgentForge task created	Verify project bearer + registration

portfolio-create — registry motion

After poc-ship and deploy audit, portfolio-create registers the slug for public proof on portfolio.nltlabs.ai. This flow is lighter than poc-ship but still crosses Engine → AgentForge → publish surfaces.

(Workflow portfolio-create.yaml not found — set NLT_ENGINE_ROOT.)

Publish checklist

Step	Owner	Outcome
Registry row	NLT Engine	FUND slug visible to operators
Site build	NLT Engine apps/portfolio	Static proof page
DNS / CDN	Cloudflare integration	portfolio.nltlabs.ai URL live
Investor link	Vol 1 portfolio story	Clickable proof in briefs

SECTION VIII

Cross-flow handoffs

How the four core sequences chain in production.

Live engine workflows at build time: create-portfolio, linear-intake-create, pe-evaluate, poc-ship, test-echo. Handoffs use git artifacts or HTTP invoke — never direct cross-repo writes.

Durable handoffs

From → To	Artifact / surface	Gate
Scout → Engine	PE JSON + idea markdown via git	intake_dispatcher
Engine → AgentForge	pe-evaluate / poc-ship invoke	FUND verdict row
AgentForge → Memory	Recall during node execution	TAPPS_BRAIN_AUTH_TOKEN
Engine → portfolio	Published slug URL	post_deploy_audit pass
Author → CI	Git diff on consumer repo	tapps_validate_changed gate

Operator trace
Start at Scout intake files, follow pe-evaluate audit in AgentForge, confirm FUND in registry, then poc-ship through deploy audit to portfolio.nltlabs.ai.

- Vol 3 maps static boundaries; this section maps motion between them.
- Vol 5 documents AgentForge executor internals referenced by every invoke.
- Vol 8 documents NLT Memory tiers behind the memory-bridge flow.

SECTION IX

Operator runbook

Day-one trace across all four core flows.

- Confirm NLT_SCOUT_ROOT and NLT_ENGINE_ROOT before trusting live tables in Vol 3.
- Trace Scout artifact → intake_dispatcher → pe-evaluate audit id in AgentForge.
- Hold poc-ship until FUND row exists; verify post_deploy_audit before sharing URL.
- Run tapps_session_start + tapps_memory search before architecture edits.
- On gate failure: quick_check locally; validate_changed with explicit file_paths in CI.

Triage index

Symptom	First check	Volume
Vol 7 stub depth	NLT_SCOUT_ROOT path	Vol 3 env table
Empty PE verdict	pe-evaluate on_error + analyst logs	Vol 4 Section II
Memory always empty	TAPPS_BRAIN_AUTH_TOKEN	Vol 4 Section IV
CI gate fail on PDF PR	build-pdfs.mjs --audit	Vol 9 ops

Confidential — prepared for NLT Labs stakeholders. Outward names follow docs/NLT_BRAND.md. Internal checkout names appear only in builder ops volumes (9–10).