NLT labs
NEW LOGIC TECH

AGENT-DNA · SPECIES III · DESIGN EDITION

Web-Store-DNA

The storefront + marketing species on AgentForge

68-gene genome · 11 chromosomes · 14 skill packs.
Evidence-first: 22-agent research pass, adversarially verified.
Operator manual: publish, run, and gate on AgentForge.

AUDIENCE Founder · Agent architects · Platform engineering

Version · v1.0 **Issued · 2026-07-28**
NLT Labs · New Logic Tech

Executive summary

Web-Store-DNA is the third Agent-DNA species: a catalog of small, single-verb agents (genes) and workflow compositions (chromosomes) that create and market web storefronts — products, services, charts, payments, stories, checkout, and multi-channel marketing — running entirely on AgentForge as a **Pattern A** consumer (HTTP self-registration) in the **Agent-DNA** archetype: gene and chromosome catalogs for one niche. Both axes are defined in the Platform Integration Map. This document is the **design**, not the as-built: the genome is specified, the repo is scaffolded and registered, and the Phase 0 build is next.

The niche is hostile in a way the first two species were not. Species one mis-files an email at worst; species three touches money, customers, statutory law, and platform enforcement systems. Four design consequences follow, each traceable to a verified evidence base:

- **More judges than producers.** Eleven judgment genes gate eleven production genes; every action class is judged, and irreversible actions sit behind closed-mandate human gates.
- **Caps live in credentials, never prompts.** Refund keys are scoped to one gene; spend allowances are enforced by the payment provider; per-channel tokens are isolated. Prompts constrain style; credentials constrain blast radius.
- **Reversibility is priced by business consequence, not API shape.** Three action classes (R / T / I) drive the rollout order — observe, draft, publish, transact, and only then irreversible acts.
- **Deterministic beats LLM wherever judgment is not needed.** KPI math, feed serializers, consent checks, and validators run as zero-cost workflow nodes.

Part IX is the operator's manual: how to leverage the AgentForge instance already wired into this repo — publish the genome, mount the skills, run the chromosomes, approve the gates, and grade the soak before each phase.

PART

Part I — The Agent-DNA program

The unit of reuse is not a monolithic agent — it is a **gene:** one agent with one verb, one object, a strict JSON I/O contract, least-privilege tool grants, and a capability tag. Chromosomes compose genes into DAGs; alleles are published versions; skills are epigenetics; the immune system reviews at publish and constrains at run time; heredity is the memory funnel.

Species lineage

Species	Niche	Status	What it proved / must prove
1 · personal-ops	A founder's life: email, calendar, files, finance	Phase 0 built	The genetics map 1:1 onto AgentForge primitives
2 · web-dev-dna	A codebase's exhaust: CI, PRs, issues, deps	Designed	Cross-species gene reuse makes species two cheaper
3 · web-store-dna	A market: storefronts, catalog, payments, marketing	Designed + scaffolded (this doc)	The genome survives an adversarial niche — law, money, platform bans

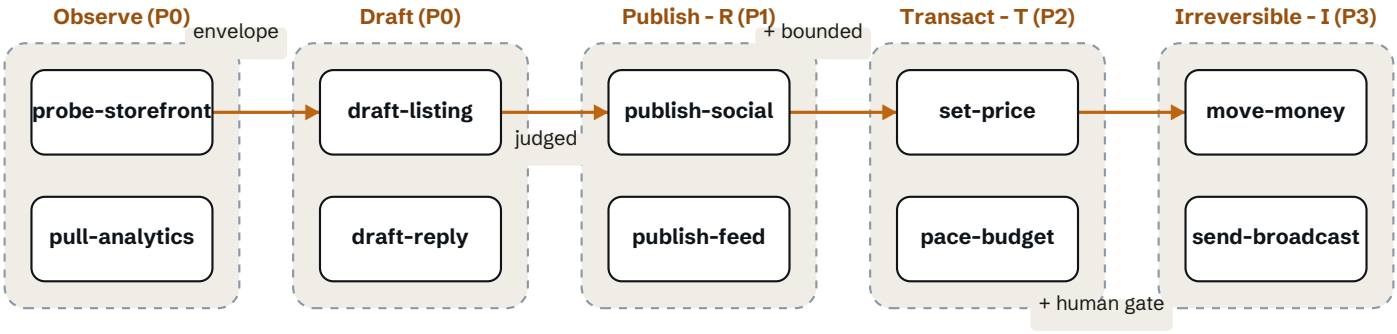
Genetics to AgentForge, 1:1

Metaphor	AgentForge primitive
Gene	One agent config file, schema 2.1, single capability {verb, object}
Allele	Published version (content-hashed, activate / rollback)
Chromosome	Workflow YAML — DAG of task / gate / python / http nodes + loops
Gene expression	Matcher + capability routing + orchestrated invoke
Epigenetics	Project SKILL.md overlays, slug-isolated, hot-reload
Immune system	Expert panel (security veto), typed guardrails, policy engine, gates
Heredity	Brain memory funnel — one auditable writer per species
Fitness	Golden cases + quality score + goal_verify — measured, never asserted

Six design rules apply to every gene, inherited verbatim: one verb one object · strict contracts · declared role · least tools · memory honesty · judged, not trusted. Nine genes re-express directly from the shared genome (classify, summarize, rank, correlate, extract, plan, notify, plus the memory curator / librarian pair) — same prompts, new epigenetics.

TRUST GRADIENT — FIVE STAGES, JUDGED AT EVERY BOUNDARY

Rollout phases (0..3) follow this gradient exactly; nothing skips a stage.



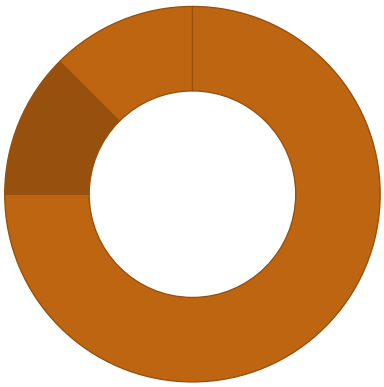
R = reversible publish (autonomous behind judges) · T = transact-adjacent (judge-gated, bounded) · I = irreversible (closed-mandate human gate).

PART

Part II — The evidence base

The species re-ran the program's evidence discipline rather than inheriting it: a 22-agent research pass (611 tool calls) swept nine areas — agentic-commerce protocols, storefront platforms, product content, social marketing, lifecycle marketing, analytics/CX, the OSS ecosystem, practitioner media, and benchmarks — plus three critic-driven gap-fills (tax/VAT, subscriptions and services, shipping and fulfillment). Load-bearing protocol claims went through adversarial verification.

Adversarial verification of load-bearing protocol claims



The one refuted claim (ACP checkout headers) was corrected from the spec YAML itself — the reason the design derives contract tests from machine-readable specs, never prose docs. The correction rate (about 25 percent) is assumed to apply to unverified areas; the research report lists every weak-sourced item honestly.

What the 2026 landscape means for the design

- **Discover in AI, buy on site.** In-chat checkout stalled (about 30 merchants live; roughly 3x worse conversion) while AI-referred traffic grew 138 percent YoY and converts 54 percent better. The species invests in feeds, UCP profiles, and agent-readable surfaces now; full ACP checkout is deferred behind named re-entry triggers.
- **Liability lands on the founder.** The AI platforms are explicitly not the merchant of record; ad tooling executes budget changes with no confirmation screen. The judge genes and credential caps are the only controls that exist.
- **The constraint surface is dense, dated, and enforced.** EU AI Act Art. 50, the FTC fake-reviews rule, TCPA statutory damages, deliverability cliffs, platform ban ladders. Compliance ships as dated, versioned skills with scheduled staleness checks.
- **APIs beat browsers roughly 2:1** on shopping-task benchmarks; reliability collapses under repetition (pass^1 61 percent to pass^8 25 percent). The namespace is API-only and acceptance is measured as pass^k.

First-target decisions (from the research decision table)

Decision	First target	Deferred / rejected
Storefront platform	Shopify (one revenue store)	Woo for stores 2..N; Wix throwaway experiments; Squarespace / BigCommerce rejected
Agentic discovery	UCP via Shopify defaults — verify, not build	—
In-chat checkout	Feed-level on-ramp only	Full ACP build waits on adoption triggers
Authorization	AP2 open/closed mandates as the gate artifact	Crypto rails out of scope
PSP	Stripe — restricted keys per gene	PayPal Agent Ready as watch item
Email / SMS	Klaviyo two-mount (read vs write)	SMS always Phase 3
Organic social order	Threads, YT Shorts (private), IG, Pinterest	TikTok inbox-drafts only (TOS)
Paid ads	Meta first, Google second	TikTok Ads deferred
Analytics	ShopifyQL (money) + GA4 (behavior) + own MER	Paid attribution platforms rejected

New lessons the research earned (L11-L18)

#	Lesson	Consequence in the genome
L11	Liability lands on the founder; platforms provide no gates	Caps in credentials (RAKs, allowances); judges are load-bearing
L12	HTTP 200 is not done — silent no-op publishing exists	Completion criteria assert server-confirmed visibility
L13	Reversibility is priced by business consequence	Three action classes; compensations are requests, never undo; 48h re-checks
L14	Compliance is dated knowledge	Skills carry expiry dates; a drift gene polices staleness
L15	Prompt injection is universal; quarantine is architecture	trust tag on every envelope item; scan gene ahead of any write credential
L16	Platform-native AI commoditizes generation	Differentiate on governance, verification, cross-channel truth
L17	Discover in AI, buy on site	Feeds and agent-readable surfaces now; checkout deferred with re-entry triggers
L18	Anti-Goodhart: judge optimizers on metrics they do not move	Pacing judged on first-party MER and refund rate, never platform ROAS

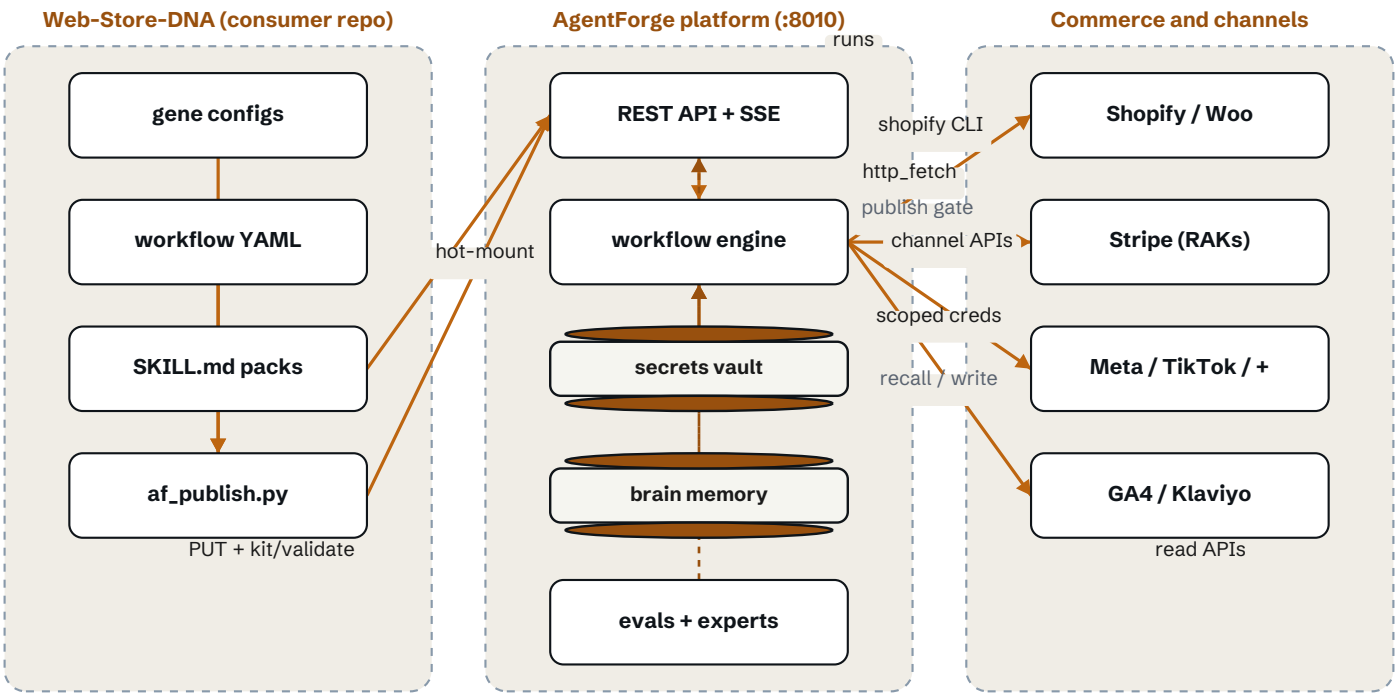
PART

Part III — System topology

Everything the species does lives in this repo as text — gene config files, workflow YAML, SKILL.md packs — and is published into the **web-store** project of a running AgentForge instance over HTTP (Pattern A). Nothing lands in the platform's backend; platform gaps are declared as dependencies (Part VIII), never worked around.

SYSTEM TOPOLOGY — CONSUMER REPO, PLATFORM, AND THE COMMERCE WORLD

Pattern A: the repo publishes genes and chromosomes into AgentForge; AgentForge executes them against the world.



Solid = primary path. Dashed = advisory or memory. All egress rides CLI grants and http_fetch allowlists closed over at spawn.

The shared envelope — one contract, trust-tagged

Every ingest gene emits the same item envelope the program established in species one, with two species-three extensions: an artifact_ref for payloads too big for body_text (images, video, feeds, evidence packs) and a **trust** field — the quarantine tag.

```
{"items": [{
"source_id", "kind", "ts", "actor", "subject",
"body_text", "attachments": [], "link",
"artifact_ref", # ADR-024 run artifact store
"trust": "merchant | customer | external"
}]}

# kind: product | order | customer | ticket | review | post |
# metric | ad_entity | creative | shipment | subscription |
```

```
# tax_event | policy_doc
```

The quarantine rule (L15): ingest genes must set trust on every item; any gene holding write or spend credentials must not consume customer or external text that has not passed the injection-scan judge. This is an architecture rule, not prompt hardening — it holds at any nonzero attack success rate, model-independent.

Anatomy of a gene

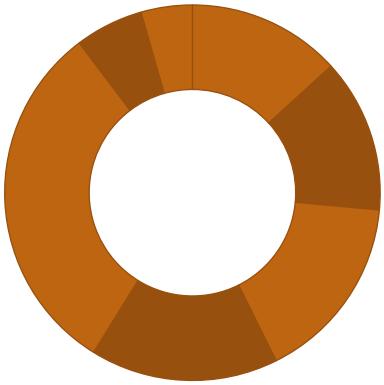
Field	What it declares
capability {verb, object}	One verb, one object — wstore.act.money, wstore.judge.claims
role	gateway / router / producer / planner / aggregator / judge
input_schema / output_schema	Strict JSON Schema; machine-checked at invoke time
allowed_tools + tool_targets	Least grant: CLI prefixes, http_fetch allowlist, or nothing
credentials	Vault key names — values never in git
guardrails	anti-secret, anti-pii, anti-mimicry, extension rules
max_budget_usd / model	Tier pinned per gene: haiku 0.10, sonnet 0.50, opus 2.00
memory_profile / footprint	readonly for all but the curator — one auditable write path
golden_cases	3+ per gene; schema-valid, guardrails-clean, one rubric
completion_criteria	Effect checks: read-back, visibility, PAUSED-state assert

PART

Part IV — The gene catalog

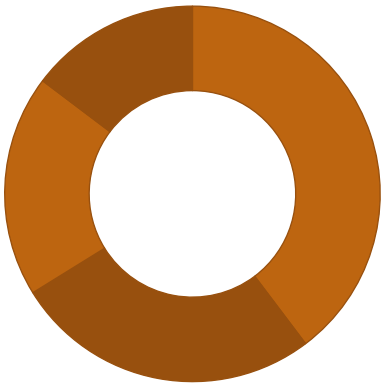
The research pass consolidated about 230 raw candidates into roughly 150 designed genes; this design curates a 68-gene **expressed genome** and leaves the rest in the gene pool — designed, evidence-tagged, unexpressed until a chromosome needs them. Entries needing no judgment ship as deterministic workflow nodes (node) at zero LLM cost.

Expressed genome by family (68 genes)



Ingest: 9
Cognition: 9
Production: 11
Judgment: 11
Action: 21
Memory: 4
Experts: 3

Genes entering service per rollout phase (of 68)



Phase 0 · observe / draft: 27
Phase 1 · publish (R): 18
Phase 2 · transact (T): 13
Phase 3 · irreversible (I): 10

Ingest family — read-only, zero write credentials

Gene	Capability	Tools / notes	Ph
probe-storefront	ingest.storefront	http_fetch: /.well-known/ucp, /api/mcp, sitemaps — own and competitor stores	0
pull-analytics	ingest.analytics	shopify CLI (ShopifyQL) + GA4 API; Shopify = money truth, GA4 = behavior truth	0
pull-adspend	ingest.adspend	Meta Ads CLI (read) + Google Ads REST, batched under rate tiers	0
fetch-social	ingest.social	IG / Threads / YT / Pinterest insights + quota and entitlement probes (L12)	0
pull-orders	ingest.order	shopify CLI + Stripe read key — orders, returns, subscriptions, disputes	0
pull-tickets	ingest.ticket	Helpdesk read API + review reads; trust: customer	0
monitor-citation	ingest.geo	Assistant-probe corpus — is the store cited by AI shopping surfaces?	0
audit-config	ingest.configdrift	Read-only platform config vs desired-state YAML in git (L13)	1
ingest-supplier	ingest.doc	doc-extract template; supplier PDFs; trust: external	2

Cognition family — pure LLM, no tools

Gene	Capability	Contract highlights	Ph
classify (shared)	cognition.classify	Taxonomy-blind; labels come entirely from skills; confidence 0.7	0
summarize (shared)	cognition.summarize	items + audience + length — summary, key points, open questions	0
rank (shared)	cognition.rank	Candidates + criteria — ordered list + scores	0
correlate (shared)	cognition.correlate	Fan-in: orders x analytics x spend x social; KPI math itself is python nodes	0
extract (shared)	cognition.extract	Claims manifest: every product fact carries a source span or is null	0
explain-anomaly	cognition.anomalycause	Hypotheses only, never actions — the one warranted LLM-judgment spot	0
plan-content	cognition.contentplan	Channel-ordered calendar with jitter (robotic timing is a ban signal)	1
plan-campaign	cognition.mediaplan	Versioned campaign-plan JSON from read-only data (draft-first)	2
plan-storefront	cognition.platformplan	Platform decision table: Shopify first, Woo 2..N, Wix throwaway	3

Production family — artifacts, never actions

Gene	Capability	Notes	Ph
draft-listing	produce.listing	Per-channel copy from sourced attributes; refuses unsourced claims	0
draft-post	produce.postcopy	Per-channel variants mandatory — byte-identical fan-out is a ban signal	0
draft-reply	produce.customerreply	Citation per claim; commitments listed (Air Canada precedent)	0
compose-digest	produce.digest	Decision-first owner digest from upstream outputs + charts	0
render-chart (node)	produce.chart	Deterministic plotting — spec in, image out	0
render-feed (node)	produce.feed	Canonical record to ACP / UCP / Merchant / marketplace serializers; versioned	1
generate-image	produce.productimage	Multi-reference conditioning; C2PA / SynthID provenance preserved; spend cap	1
render-video	produce.shortform	HeyGen / Runway + C2PA; hardcoded refusal of testimonial personas (FTC 465)	1
draft-flow	produce.lifecycleflow	Flows + segments from margin data the ESP cannot see (L16)	1
draft-adcreative	produce.adcreative	Variants under a similarity ceiling; feeds the fatigue loop	2
draft-taxdossier	produce.taxdossier	Registration packet that names but never holds PII; fails closed	2

Judgment family — the densest family, by design

Gene	Capability	Notes	Ph
scan-injectionpayload	judge.quarantine	Deterministic parser + ruleset over all non-merchant text; fail closed	0
judge-claims	judge.claims	Claims-manifest validator + regulated-term lexicon; fail closed	0
judge-disclosure	judge.aidisclosure	Per-channel AI-disclosure decision table + C2PA verify — never one flag	0
judge-originality	judge.originality	Thinness / near-duplicate screen (slop enforcement is live on Google / YT)	0
judge-reply	judge.replysafety	Uncited claim / commitment / disclosure / PII rejection; fail closed	0
validate-feed (node)	judge.feedvalid	Pinned-schema validators derived from spec YAML, never prose	1
judge-sendrisk	judge.sendgate	Deliverability (DNS, complaint rate) + deterministic consent-provenance check	1
judge-spend	judge.spendaction	Policy constants as data; allowance mirror-check before any network call	2
judge-compliance	judge.regulated	Routed rubric over the dated legal skills (ARL, reviews, pricing, EU)	2
judge-launch	judge.storelaunch	Go-live evidence bundle: checkout sim, KYC, tax, policies	3
judge-visual	judge.visual	Own-storefront screenshots; image half available today via AF vision-QA gate	3

Action family — three reversibility classes (L13)

Class R — reversible publish (Phase 1). Autonomous behind judges. Extension guardrails on all: API-only, quota-ledger preconditions, per-channel credential isolation, server-confirmed visibility.

Gene	Wiring
notify	Deterministic webhook template (Phase 0 — every digest ends here)
upsert-product	shopify store execute with per-mutation allowlist; creates as DRAFT; full list sets
publish-feed	Feed push + /.well-known/ucp deploy; refuses without validate pass; versioned rollback
publish-social	Channel order: Threads, YT Shorts (private), IG story, IG post, Pinterest; TikTok = inbox drafts only
stage-email	ESP write grant, DRAFT / cancellable states only
publish-reviewreply	Review-platform reply endpoint; sentiment-blind solicitation (FTC 465)
emit-conversion (node)	CAPI / GA4 with order-id dedup — idempotent

Class T — transact-adjacent (Phase 2). Judge-gated and bounded: append-only audit logs, idempotency keys everywhere, delta envelopes.

Gene	Wiring
set-price	Input-provenance whitelist in the contract (pricing statutes); margin floor; audit log
configure-store	Scoped shipping / market / checkout mutations, branched on shipping-model audit
pace-budget	Max 20 percent delta inside a pre-authorized band; pause is the only ungated spend action
fulfill-order	Explicit line items mandatory; inventory compare-and-set never disabled
send-reply	Helpdesk send on an intent allowlist; money intents always escalate
sync-subscription	Payment links + cancel-at-period-end class only; statutory renewal notices
record-taxevent	Tax transactions exactly-once, keyed on capture

Class I — irreversible (Phase 3). Human gate mandatory. Every gate payload is an AP2-shaped mandate: open mandate = the autonomous draft, closed mandate = the human-signed, hash-pinned approval.

Gene	Wiring
launch-storefront	Go-live composite; requires the judge-launch evidence bundle; store provisioning is a human task
activate-campaign	PAUSED-to-ACTIVE flip only; lifetime cap; kill-switch sibling; 48h re-check scheduled
send-broadcast	Recipient count, segment, consent basis, deliverability verdict in the gate payload; seed list first
move-money	Capture / refund / payout via per-purpose restricted keys held by no other gene
buy-label	Local dedup ledger (carriers ship no idempotency); void-label is best-effort, not undo
file-tax	Registration + filing as tracked human tasks; collection hard-blocked on permit evidence
archive-product	Archive default; delete-class mutations isolated here; impact report + redirects first

Memory and heredity + project experts

Gene / expert	Role
memory-curator (shared)	Sole writer to the brain namespace; distills quota incidents, API quirks, brand decisions
kb-librarian (shared)	Cache-first recall opening every chromosome; failure memories surface as KNOWN ISSUES
keep-evidence	Persists order-time mandates, signatures, consent — the dispute-evidence pack (L11)
audit-staleness	Scheduled checks of dated skills against spec releases and regulation feeds (L14)
expert-commerce	Merchandising, pricing psychology, conversion — critic in listing and pricing loops
expert-brand	Voice, story arcs, creative direction — critic in content loops
expert-compliance	The dated legal surface; advisory only, non-authoritative for legal and tax

PART

Part V — Skills (epigenetics)

Tuning behaviour is a skill edit (hot reload), not a republish. The species-three rule (L14): **every skill is a dated, versioned pack with an explicit re-verify date**; the staleness gene polices expiry. The research pass designed 81 packs; ten are mounted in dna-core today.

Skill	Tier	Injected into	Content
passk-eval	core	golden authoring	pass^k acceptance thresholds by gene role
ai-disclosure-matrix	core	judges, render genes	Per-channel disclosure table + C2PA preservation; the dated pack
kpi-truth	domain	correlate, digest, nodes	Formula sheet, source-of-truth routing, MER objective (L18)
cx-guardrails	domain	reply pipeline	Assertable vs never-assertable; escalation triggers
site-compliance	domain	page draft + audit	Privacy, consent, accessibility, price display — dated, expires 2026-10-01
agentic-surface	domain	probe / feed / publish	UCP profile, MCP cart/checkout endpoints, agentic-catalog eligibility
quarantine-rules	species	scan-injectionpayload	Attacker-surface table bound to seven named ingest genes
taxonomies	species	classify	Ticket / review / anomaly label sets; classify is taxonomy-blind
store-atlas	deployment	planners, all action genes	Registry of truth per storefront; unlisted target does not exist
brand-voice-pack	deployment	draft + judge genes	Voice, banned phrases, proof points; brand fields UNSET until Phase 1

Reuse tiers — what travels to the next species

The tier on each pack is the portability contract, and `scripts/validate.py` enforces it so a pack cannot drift out of its tier unnoticed. This is what makes the genome reusable rather than one-off: `render_dna_core.py --export` scaffolds a new species from the core and domain tiers with every species-owned slot left blank, and `--instance` stamps a new store of this species with the species content re-slugged to the new identity.

Tier	Who may take it	Enforced invariant
core	any species, verbatim	No coupling to this gene catalog or to commerce; exported
domain	any commerce species, verbatim	No reference to a gene in this catalog; exported
species	travels only with a rewrite	Names genes in this catalog; not exported
deployment	populated per storefront	Ships UNSET; only deployment packs may contain UNSET

Two packs are dated rather than evergreen — ai-disclosure-matrix and site-compliance — so a judge reading either past its expiry must say so in its findings rather than silently trusting stale thresholds.

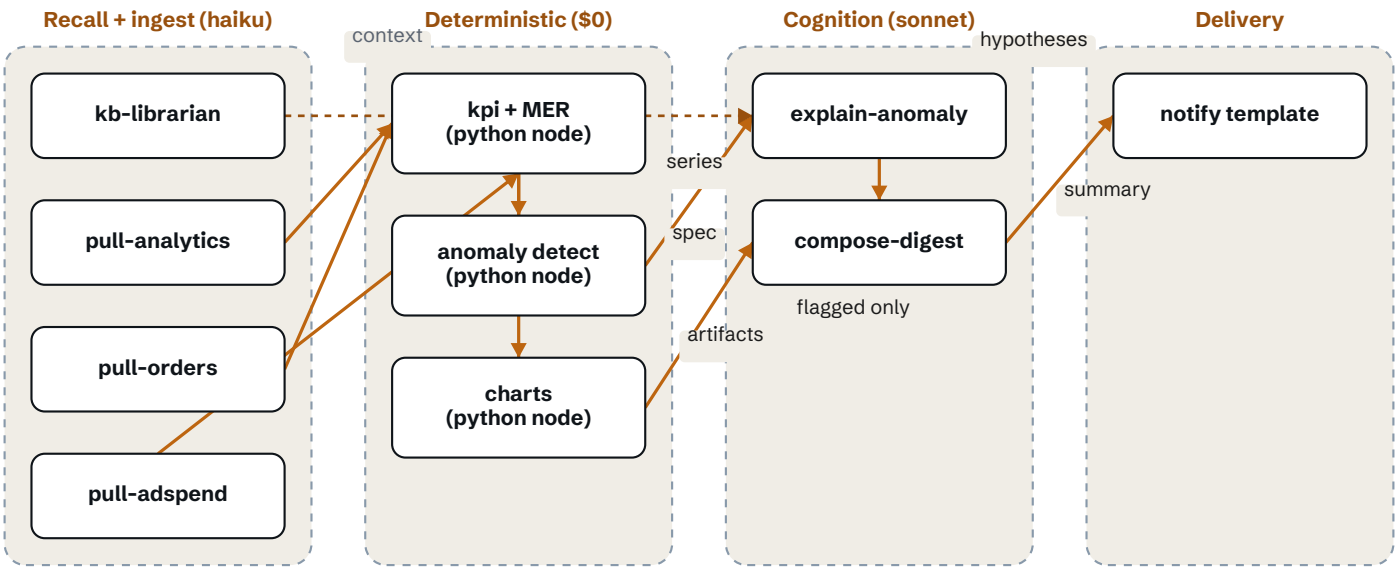
PART

Part VI — Chromosomes (the workflow catalog)

Conventions inherited from the program: on_error partial, a top-level goal with a verify predicate, output schemas on task nodes, gate timeouts, publish via PUT and validate via kit/validate with dry-run. Every cadenced chromosome self-skips on the quiet path; every chromosome states its Phase 0 reduction so the doc and the repo cannot drift apart silently.

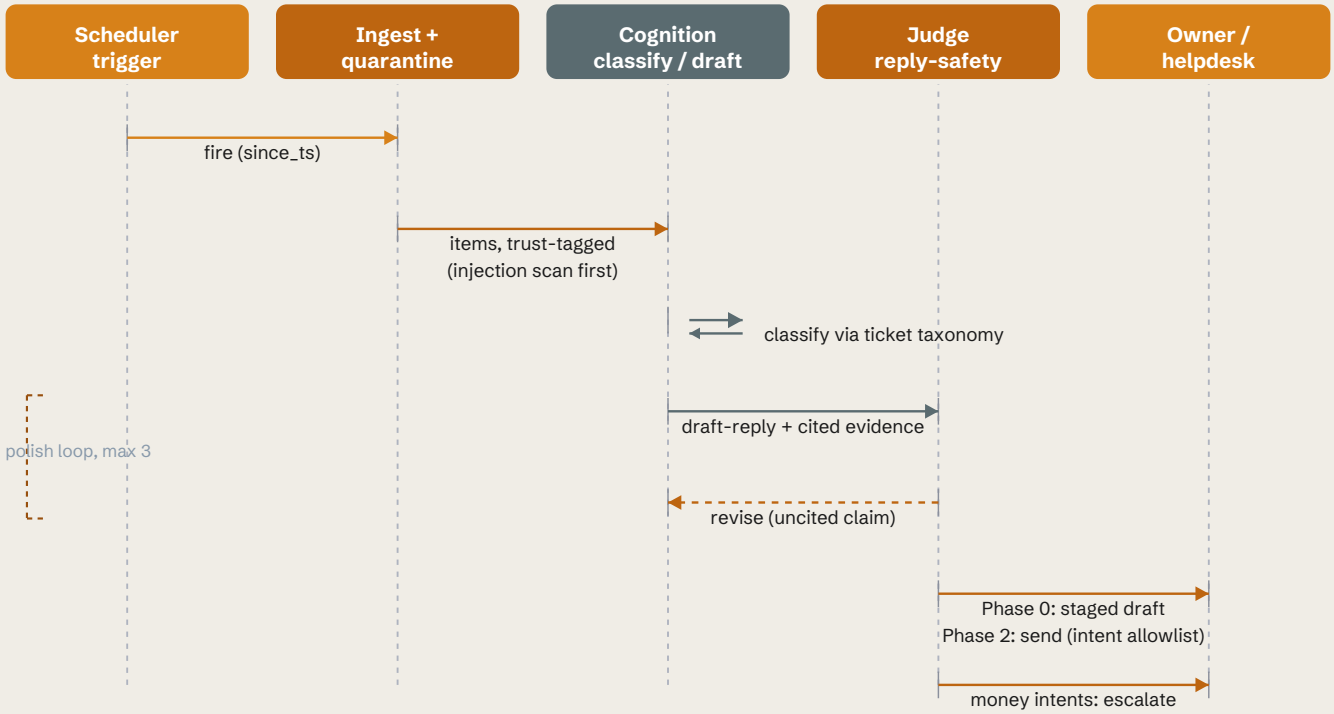
CHROMOSOME — STORE-HEALTH (DAILY DIGEST DAG)

Three of nine nodes are $0 deterministic nodes; the quiet path costs almost nothing on a normal day.



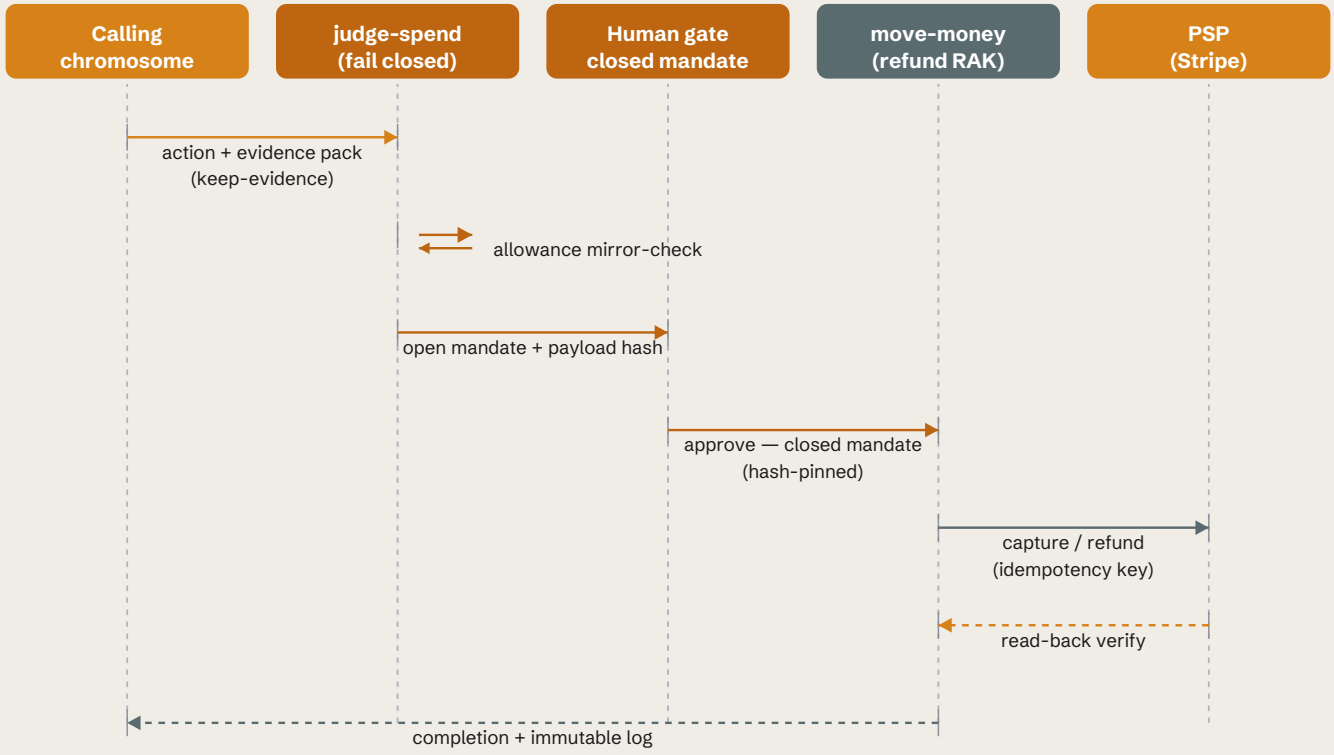
Quiet path: no anomalies means explain-anomaly self-skips — the cadence costs pennies when nothing happened.

CHROMOSOME — cx-inbox (every 30 min; draft-only in Phase 0)



Every reply carries per-claim citations (Moffatt v. Air Canada) and the first-turn AI disclosure (EU AI Act Art. 50). Fail-closed judge.

CHROMOSOME — money-movement (the only path that moves money)



Refunds below the (default-zero) ceiling may skip the gate — never the judge or the read-back. Expired tokens fail; they are never silently retried.

The full catalog

Chromosome	Cadence	Phase 0 reduction	Full
store-health	daily	none — ships whole	0
agent-readiness	weekly	none — probe + citation + digest	0-1
listing-pipeline	per batch	stops at judged drafts in the artifact store	0-1
content-calendar	daily	plan + draft + judge to digest; nothing posts	0-1
cx-inbox	30 min	draft-only; replies staged for the owner	0-2
ad-operations	daily	observe + plan + digest; 72h observation on connect	0-3
lifecycle-flows	weekly	flow designs in the digest	1-3
order-fulfillment	30 min poll	order and shipment digest only	2-3
money-movement	on demand	— (Phase 3 by definition)	3
store-bootstrap	on demand	designed now, assembled after the soaks	1-3
compliance-watch	weekly	staleness + config-drift digest	1-3

> “Every optimizing gene is judged on a downstream metric it does not move — the canonical failure is a 31-day autonomous ads run that hit its CPL target on paper while lead quality collapsed.”
>
> **— Lesson L18 — anti-Goodhart, from the practitioner evidence**

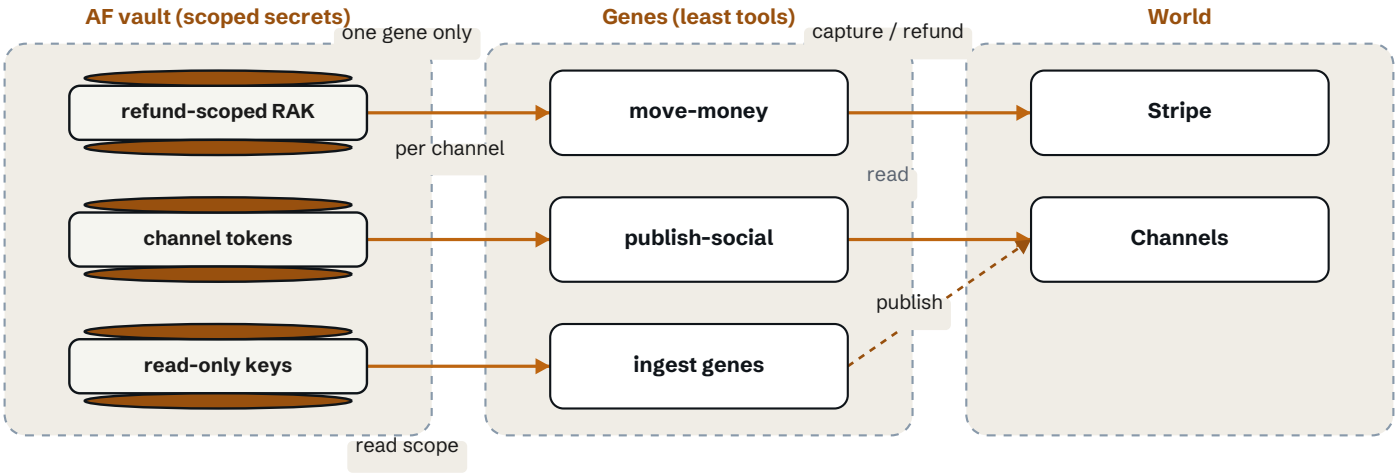
PART

Part VII — The immune system

Risk tiers: ingest and cognition low / medium; Class R medium; Class T and I high — the validator then requires guardrails and completion criteria. Nothing is trusted. Every gene carries anti-secret and anti-pii; production genes add anti-mimicry (agent attribution on public replies) and the hardcoded anti-testimonial rule; action genes add per-class extension rules.

CREDENTIAL CUSTODY — CAPS LIVE IN CREDENTIALS, NEVER PROMPTS

Blast radius is bounded below the agent layer: per-purpose keys, scoped to single genes.



Platform ask PD-10 (af-dna-platform-evolution-2) turns the agent-scope from convention into an enforced guarantee.

Human gates — more than either prior species, each evidence-priced

Gate	Why (evidence)
Go-live / checkout enablement	KYC can fail and freeze payouts; the browsable-to-buyable boundary
Campaign activation + budget-band escalation	No vendor confirmation screens exist; documented runaway spend
Broadcast and every SMS send	0.30 percent complaint rate = permanent rejection; TCPA statutory damages per message
Money movement (capture / refund above ceiling / payout)	Merchant-of-record liability is the founder's (CONFIRMED)
Label spend	Carriers ship no idempotency; refunds are best-effort and rejectable
Tax registration and filing	Legal attestation; collecting without a permit is illegal
TikTok direct publish	Platform TOS requires human preview and manual privacy selection
Subscription immediate-kill; product delete	Mandate destroyed / listing history unrecoverable

Budget as selection pressure

Tier	max_budget_usd	Where
haiku	0.10	Ingest, classify, digest — the cadenced bulk
sonnet	0.50	Cognition, production, judges
opus	2.00	draft-theme only (storefront code generation)
deterministic nodes	0.00	KPI math, validators, serializers, consent checks

External-API render genes (image, video) carry an additional spend cap; ad and marketing spend is not an LLM budget and is governed by the Class T / I envelopes and gates above.

Now enforced, not aspirational. The platform activated its policy engine and added workflow-context and capability predicates on 2026-07-28, so the DENY rules above are written and live at policy/web-store.rules.yaml — including phase-gate denies, so invoking a Phase 1 or 2 capability before its phase is promoted is a policy event rather than a discovery. Credential custody, fail-closed judges, and load-time coherence gates sit beneath it as defense in depth. Evals: every gene ships 3+ golden cases with seeded traps (an injection payload in a product description, an unsourced claim, an unauthorized refund promise, a fabricated testimonial), and acceptance is pass^k — now a first-class platform field, used from day one.

PART

Part VIII — Platform dependency ledger

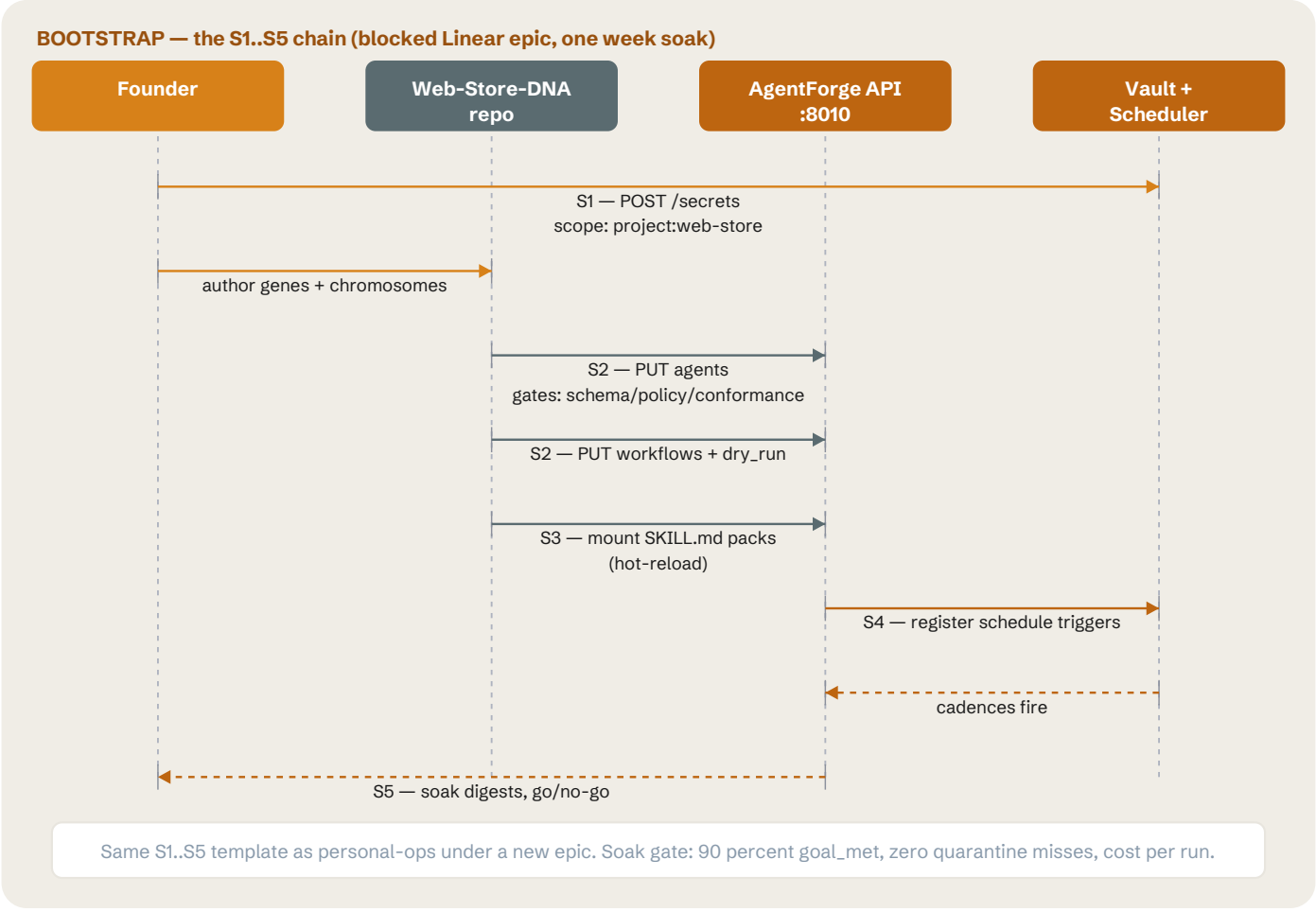
Phases 0-2 need zero platform changes, and the platform moved further than that: the credentials, policy, ledger, MCP-least-privilege, and pass^k asks all shipped on 2026-07-28, so several guardrails the design called aspirational are now enforced. One new gap was found in the other direction — a consumer project has no zero-cost compute node. The full ask set lives in af-dna-platform-evolution-2.md in the AgentForge repo.

Capability needed	Platform state (2026-07-28)	Gates
CLI + http_fetch ingest / publish surfaces	Shipped — tool_targets, allowlists closed at spawn	Phase 0 runs now
Owner notification and digests	Shipped — notify template, artifact store	Phase 0
Policy denies by workflow context / capability class	SHIPPED (PD-9) — engine activated, both predicates added	Phase 0 — rules written and live
Per-gene credential custody + expiry / usage caps	SHIPPED (PD-10) — agent scope enforced at resolution	Phase 0 authorable
MCP least privilege — vault creds, per-server tool subsets	SHIPPED (PD-11)	Phase 1 (ESP two-mount no longer waits on PD-1)
Durable ledgers (quota, dedup, consent)	SHIPPED (PD-12) — project_ledger + kind: ledger node	Phase 1
pass^k golden acceptance	SHIPPED (PD-15a) — trials + pass_threshold	Phase 0 — used from day one
Consumer-usable deterministic compute (the $0 node path)	NOT AVAILABLE — python resolves backend.* only; script needs plugin registration	New ask PD-19; Phase 0 derives nothing
Order webhooks / event-fired chromosomes	Event triggers shipped; one-shot 'once' in flight	Phase 2 (PD-13)
Video artifacts above 25 / 100 MB caps	Object-store backend still a seam	Phase 1 video (PD-4)
Own-output browser verification	No preview surface; vision-QA ship gate covers image artifacts	Phase 3 (PD-3), narrow by design (L10)

PART

Part IX — Leverage, build, and run with AgentForge

This repo is already wired: the integration kit is installed (eleven af-* skills for Claude Code and Cursor), the agentforge MCP server points at the local instance on :8010, the **web-store** project is registered, and NLT Quality Pipeline provides quality gates and shared memory (brain namespace pinned to web-store). What follows is the operating manual for taking the genome from this document to a soaked, running Phase 0.



1 · Verify the platform and the wiring

```
# health (this repo pins :8010)
curl -s http://localhost:8010/health | jq .status

# the project is registered (id ca96593e...)
curl -s http://localhost:8010/projects | jq '.[] | select(.slug=="web-store")'

# kit verification + doctor
bash $AGENTFORGE_REPO/clients/agentforge-integration-kit/scripts/af_kit_install_verify.sh
--repo-root .
uv run python $AF_KIT/scripts/af_doctor.py --slug web-store --repo-root .
```

2 · The Phase 0 slice is authored — validate it offline

The slice is in the repo: **25 genes** (7 ingest, 6 cognition, 4 production, 5 judgment, notify, 2 memory), **5 chromosomes**, **8 skill packs**, and the policy rules file. A standalone validator checks the whole kit without a running platform — frontmatter shape, schema JSON, golden-case and pass^k bounds, workflow wiring, unknown agent references, and node kinds a consumer cannot actually use:

```
$ uv run python scripts/validate.py
checked 25 agents, 5 workflows, 8 skills, 1 policy file(s)
kit is publishable
```

Genes live under agentforge/projects/web-store/agents/, one config file per gene directory; chromosomes under the sibling workflows/ directory. The sketch below shows the shape of an action gene.

```
# agentforge/projects/web-store/agents/wstore-notify/the agent guide (sketch)
schema_version: "2.1"
capability: { verb: act, object: notify }
role: gateway
template: notify # deterministic — no LLM
risk_level: medium
credentials: [WSTORE_NOTIFY_WEBHOOK]
guardrails: [{ type: anti-pii }, { type: anti-mimicry, target: owner }]
completion_criteria: webhook 2xx read-back
golden_cases: 3 # schema-valid, guardrails-clean, one rubric
```

3 · Vault the credentials (S1)

```
# scoped to the project — values never touch git
curl -sX POST http://localhost:8010/secrets \
-H 'Content-Type: application/json' \
-d '{"key": "WSTORE_SHOPIFY_CLI_TOKEN", "value": "...",
"scope": "project:web-store"}'

# Phase 0 set: Shopify read token (PCD approval has lead time),
# GA4 + social read tokens, notify webhook. Write keys arrive with
# their phases — the refund-scoped RAK not before Phase 3.
```

4 · Publish, validate, activate (S2)

```
# the kit publish flow (or the af-publish skill from Claude Code)
uv run python $AF_KIT/scripts/af_publish.py --slug web-store --layout scout

# which drives, per gene:
# PUT /projects/web-store/agents/{gene} -> v1 auto-active
# gates: schema -> policy -> conformance -> cohesion -> expert panel
# then the kit as a whole:
# POST /projects/web-store/kit/validate -> verdict envelope
# PUT /projects/web-store/workflows/{name} + workflow_dry_run
```

5 · Mount skills (S3) and schedules (S4)

```
# skills are files — hot-reload, slug-isolated
$AF_EXTERNAL_SKILLS_ROOT/web-store/<skill>/SKILL.md

# cadences are declarative triggers on the published workflows:
# store-health: cron 0 7 * * *
# cx-inbox: interval 1800s
# agent-readiness + compliance-watch: weekly
# missed fires self-heal on the next cadence — nothing here is
# cadence-critical to the minute.
```

6 · Run it — invoke, watch, approve

```
# ad-hoc goals route through the matcher / orchestrator
curl -sX POST http://localhost:8010/projects/web-store/tasks/invoke \
-d '{"prompt": "why did revenue dip Tuesday?", "orchestrate": true}'

# run a chromosome directly (dry_run first is the habit)
curl -sX POST .../projects/web-store/workflows/store-health/run \
-d '{"inputs": {"since_ts": "2026-07-27T00:00:00Z"}, "dry_run": true}'

# watch a run stream (SSE), then approve a pending gate with the
# hash-pinned payload — the closed mandate:
GET /workflows/runs/{run_id}/events
POST /workflows/runs/{run_id}/gates/{node}/decision
{"approved": true, "reasons": ["..."], "approver": "founder"}
```

From inside Claude Code or Cursor, the same surface is available as MCP runtime tools — health(), list_workflows(), invoke_task(), run_workflow() with dry_run, and get_workflow_run() for per-node status and cost — so day-to-day operation never requires hand-written curl.

7 · Day-2 operation

• **Digests are the product.** store-health lands every morning; cx-inbox staged drafts and agent-readiness / compliance-watch roll into it weekly.

• **Quality is watched, not assumed.** A per-gene quality composite; golden cases re-run on promote; spend ledger reviewed in the weekly rollup; pass^k acceptance before any phase promotion.

• **Gates stay ergonomic for one founder.** Batched approvals, diff previews, a standing kill switch — the human-gate-placement skill is the contract.

• **Memory compounds.** The curator distills quota incidents, platform quirks, and brand decisions; the librarian replays them into every chromosome as KNOWN ISSUES.

8 · Rollout gates

Phase	Ships	Success gate
0 — observe & draft	~20-gene slice; store-health, cx-inbox (draft), listing + content (draft), agent-readiness	1-week soak: 90 percent goal_met; reconciliation in band; zero quarantine misses; cost per run
1 — publish (R)	Feeds, UCP profile, Threads / Shorts / IG, staged flows, review replies	2 weeks, zero platform flags; visibility confirmed on 100 percent of publishes; owner keeps 80 percent of drafts
2 — transact (T)	Pricing, store config, bounded pacing, fulfillment, CX sends, tax events	pass^8 on money-adjacent goldens; zero out-of-band mutations; ad accounts healthy
3 — irreversible (I)	store-bootstrap end-to-end, campaign activation, broadcasts, money-movement, labels, tax filings	A storefront launched through the full gate chain; zero un-gated Class I actions; dispute pack per agentic order
4 — hardening	Eval mode block, provenance signing, quality floors, pool retire sweep	Program standard

PART

Appendix — Document set and repo map

Artifact	Where	What it holds
web-store-dna.md	docs/designs/	The species genome — the canonical design this PDF renders
storefront-agent-research.md	docs/research/	Evidence base: 12 areas, 150-gene candidate catalog, 81 skills, risk register, source index
af-dna-platform-evolution-2.md	AgentForge docs/designs/	Platform asks PD-9..18 + PD-1..8 status, code-verified
agentforge/projects/web-store/	this repo	Publish dirs: agents/ (genes) and workflows/ (chromosomes)
Integration contract	repo root	.mcp.json + repo agent guide — agentforge MCP on :8010, NLT tapps servers
[project config]	repo root	Kit skills: connect, author-agent, publish, workflow, smoke-test, doctor
reports/web_store_architecture/	this repo	This document's source (report-studio consumer, NLT brand)

Reading order

This PDF for the shape → web-store-dna.md for every table in full → storefront-agent-research.md when a design choice needs its evidence → the platform-evolution series when a dependency needs its file:line proof.

Web-Store-DNA — Architecture & Design · v1.0 · issued 2026-07-28 · design edition (not as-built). Diagrams are vector-native ReportLab, built with NLT Report Studio (brand nlt-v3.2). Every claim traces to the document set above.