

# Q7 ENHANCED PRECISION CALIBRATION — PIPELINE ARCHITECTURE

A LIVING MAP · 8 STAGE-GROUP LANES · AGENT-LED STRUCTURAL-INTEGRITY CORNERSTONE (S0) · STAGE-GATED S0-S10 · COVERAGE RULES R1-R8 + R-SI · ≈1000-TRIAL CONVERGENCE WITHIN A ≤1000 CAP · HAI PER-CONFIG IN-PIPELINE

2026-06-25 · WHITE-BG / PRINT EDITION · AS-BUILT: ALL 6 ALGOS RELAUNCHED ON THE AGENT-LED STRUCTURAL-INTEGRITY CORNERSTONE · STRUCTURALLY-GREEN · ≤1000-CAPPED · DD-UTIL BOOSTER HARD-WIRED · HAI-ON

Calibrate the system you actually trade — full population, every parameter, HAI-gated, per-config · **structurally-perfect before trial #1** (a clean model converges within ~1000 trials) · golden-source law · honest charter gate (DR-1) · fail-closed at every gate

## WHAT THE PIPELINE DOES

Proves the Pine + Python + pipeline structurally perfect before trial #1 (agent-led cornerstone), then calibrates 100% of the calibratable Pine population per-config — converging within a **≤1000-trial cap** — trains the per-config HAI gate in-pipeline (HAI-ON), and ships only configs that pass an honest charter gate and an independent re-derivation.

## HOW IT STAYS HONEST

An agent-led Structural-Integrity Cornerstone proves the model defect-free (8 checks + a runtime ship-smoke) before any trial, so a broken/zero-trade cycle never reaches calibration. The charter gate scores the **HAI-gated** system — the one that trades live. No off / lite / reduced mode (R1).

## HOW IT STAYS IN-SYNC

The HAI model that gates live trades is cycle-pinned (RX##+SHA) and SHA-identical to the one calibrated this cycle (R8). The HAI-Sync Validator re-derives the HAI-gated result on a holdout and gates every promotion.

## HOW IT STAYS ACCOUNTABLE

Every cycle emits a trial-ledger (per-config trial counts), a Heat Map row, a model SHA in the dossier, and an independent Validator sign-off. Flow oversees; lanes self-govern. Golden source = highest Pine v6 RX## per algo.

## BINDING COVERAGE RULES R1-R8

### R1 - 100% coverage

Every calibratable Pine param in the search space, per-config. One mode — no off / lite / reduced.

### R2 - Pine authority

Population = all `input.*` in the golden source. Pine wins over matrix/cache.

### R3 - Never cut pop

No top-K, freezing, sampling, silent narrowing. High-dim = more trials, never fewer dims.

### R4 - Signed exclusions

Only non-calibratable cosmetic/op inputs, on the USER-approved list.

### R5 - Pre-run gate

Fail-closed: confirm 100% live coverage + HAI active before any cycle, else ABORT.

### R6 - No hardcoding

No calibratable param at a literal / frozen default. Hardcode guard + pre-commit hook.

### R7 - Concerted search

Search runs to convergence within a `≤1000 total_eval_cap` (≈973 GP + startup/jitter). A clean model converges here; a BLOK after a truncated search is invalid.

### R8 - HAI per-config, HAI-ON

HAI trained per-config in-pipeline, HAI ACTIVE, cycle-pinned train=serve. No model = **BLOK**, never HAI-off. No exceptions.

## LEGEND

### GATE OUTCOME

- PASS → advance
- Gated / sign-off
- BLOK / ABORT

### STAGE-GATED FLOW

- S0 - pre-cycle null flow
- S1 - population enum
- S2 - data cert
- S3 - concerted search
- S4 - HAI tandem (R8)
- S5 - WF / CPCV
- S7 - honest charter gate
- S8-S10 - pkg + sign-off

### 7 SWIMLANES (GOLDEN SOURCE)

- S1 Perps - RX85
- S2 Bond Spd - RX07 5-LEG
- S3 Index Micros - RX33
- S4 Commod Micros - RX33
- S5 Commod Spot - RX37
- S6 FX Futures - RX02 5-LEG
- S7 Pred-Mkts - census 1168

## 0 - STRUCTURAL-INTEGRITY CORNERSTONE

AGENT-LED · FAIL-CLOSED · BEFORE TRIAL #1

### THE NARRATIVE - SOUL

"Structurally perfect before trial #1." An agent-led cornerstone proves the Pine + Python + pipeline have zero structural defects BEFORE a cycle runs — because a clean model converges within ~1000 trials, while a defective one burns thousands and ships nothing (proven: S5 finalized BLOK after 1470 trials with engine-attribution broken). The >1000-trial problem is a symptom; this kills the cause.

Enforces **R5 · R8 · R-SI**.

### FRAMEWORK - GATES

- **G0 - AGENT-LED (fail-closed):** the `q7-structural-integrity` agent runs the gate, reasons every finding, fixes + RE-PROVES GREEN, signs the cert under its own key, zero promotion authority.
- **8 checks:** C1 pine\_init · C2 entry\_present · C3 calibration\_grid · C4 searched\_param\_used (no dead dims) · C5 callback\_parity · C6 engine\_attribution (the S5 all-BLOK cause) · C7 booster\_hardwired · C8 formula\_footprint\_parity vs the algo's OWN golden Pine. `ALGO_PROFILES` = judged vs its OWN design (adapt, don't clone).

### FRAMEWORK - AGENTS

- **FORMULA-FAITHFUL, not presence (2026-06-30 hardening):** a structural cert that verifies only that parameters are DECLARED & CONSUMED can still pass a **STRIPPED SHADOW** — a Python core implementing a subset of the golden Pine's exits/SL/governance/sizing/engines (subset-implementation preserves the input signature). C8 must therefore compare *formulas function-by-function* against the algo's OWN golden Pine — exit-ladder cardinality, the SL-governance layer (lev-aligned cap · noise floor · lev-reduction), the sizing model, engine entry conditions — NOT input coverage, and NOT against an S1 template. Presence-GREEN is necessary but not sufficient for fidelity.

- **Remediation = WHOLE-FILE faithful re-port (no snippet patches):** a file with structural drift is rebuilt complete and deployable to the golden Pine; the producer STAGES, an independent verifier RE-AUDITS (producer ≠ verifier) / no calibration numbers are authored (they come from Optuna / the signed prior). Iterative snippet-patching causes cascading defects and is rejected.
- **Mandatory runtime ship-smoke:** static GREEN is NOT enough — the model must import, trade, and SHIP engines ( `Live ≥ 1`, not BLOK) end-to-end.
- **Pipeline integrity:** `≤1000` trial cap wired · `q7_precision` version-consistent · per-version (never cross-deploy) · gate-chain not blanket-overridden
- **G0 · R5 coverage + R8 HAI assertion ( `i_useHAIInBacktest` ) ON + per-config model) · null-cycle (RX86) HALT**
- `run_precycle_integrity.sh` : launch ONLY on GREEN + agent\_signed + sha-match + runtime\_finalize\_ship. No blanket override — a waiver is per-defect + human-signed.

### TECH STACK

`q7-structural-integrity (agent)` | `structural_integrity_gate.py` (C1-C8) | `run_precycle_integrity.sh` | `coverage_gate.py` | `pine_population.py` | `core_integrity.py` | `risk_control_verify`

### ASSET LIBRARY

- **Structural-integrity cert** — sha-pinned, agent-signed, with `runtime_finalize_ship` (GREEN gates the cycle)
- `q7-structural-integrity_md` agent charter
- `structural_integrity_gate.py` + ALGO\_PROFILES
- `precycle_gate.py` + tests (6/6, HALTs real RX86)
- `calibratable_population.json` + coverage manifest
- `sem_guard.sh` watchdog
- **Pre-Calibration Checklist** — the fail-closed pre-mortem control (Pine+Python+C# parity, build integrity, data, HAI readiness, gate/ship integrity, plumbing, post-run cert)
- **Pine Risk Control Catalog + verification** — every risk control (drawdown · ELU-halt · leverage/sizing · stop-loss/exit · entry gates · liquidation) cataloged from the golden Pine and verified present-active-correct in the Python/C#
- **Clean Scripts** — the cycle's golden Pine + clean Python core (+ NinjaScripts), version-stamped, with embedded-fix headers
- **Per-run Pre-Cycle Documentation** folder (`C8_Calibration_Production/Pre-Calibration` `Checklist/$fn`) · Pre-Cycle Documentation `MM-DD-YYYY` — the 4-doc set + structural-defect cert, produced before every run

### GUARDRAILS

- Fail-closed: FAIL / UNKNOWN / absent artifact ⇒ terminal
- Agent-led, no shortcuts: never GREEN with a FAIL open; fixes re-proven, not assumed
- No blanket override — a waiver is per-defect + human-signed
- **Preventive risk by construction (verified at S0):** drawdown is bounded by `SIZING`, not a reactive kill (which an overnight gap defeats). `Size n = min(n_pre, [(equity - truePeak · (1-D)) / r_worst])`; block entry when `equity ≤ truePeak · (1-D)`. `truePeak` ratchets up-only (never reset); `r_worst` = worst-case per-unit loss (stop + per-instrument gap reserve); the hard ceiling `D` is a STRUCTURAL CONSTANT, never calibrated, identical in Pine/Python/C#. The cornerstone verifies it present-active-correct on all three.
- Per-version packages: never cross-deploy a producer/parity

## 1 - POPULATION & DATA

PINE ENUM + CERTIFIED SNAPSHOT

### THE NARRATIVE - SOUL

"The Pine is the law." Enumerate the full calibratable population from the golden-source Pine (per-config), minus signed-off exclusions. Certify the data once, immutably.

### FRAMEWORK - PROCESS

- S1 - parse `input.*` → per-config dims
- Subtract R4 signed-off exclusions only
- S2 - SHA256 snapshot · leak-check · embargo

### TECH STACK

`pine_population.py` | `data_curator` | `feature_stationarity`

### ASSET LIBRARY

- Golden-source registry (RX85/RX07/RX33/RX33/RX37/RX02)
- `data_snapshot.parquet` + SHA256
- R4 signed-off exclusion list

### GUARDRAILS

- Matrix/cache never the authority — Pine wins
- Nothing excluded by assumption
- RE-XX defined in Pine first, then Python parity

### TECH STACK

`gp_search` (pluggable) | `Optuna TPE/CMA-ES` | `trial_ledger.py` | `cpvc_runner`

### ASSET LIBRARY

- optimizer backend `71004f8`
- `trial_ledger_manifest.schema.json` (contract v1)
- parquet ledger (pyarrow)

### GUARDRAILS

- Full population — no top-K / "high-impact only" / freezing
- Every calibratable param searched (hardcode guard + hook)
- Converge within `≤1000` total evals · cap is per-version (never cross-deployed)
- Booster hard-wired ON — never env-gated off (return-within-budget)

## 2 - CONCERTED SEARCH

≥1000 TRIALS · HIGH-DIM

### THE NARRATIVE - SOUL

"Converge within budget — clean models don't need thousands." Every calibratable dim is a live, optimized dimension. High dimensionality is answered with a proper optimizer and a return-within-budget objective — never fewer dims — bounded by a `≤1000` total\_eval cap so search converges instead of multiplying.

### FRAMEWORK - PROCESS

**STEP** | **WHAT**

**Optimizer** | pluggable Optuna · TPE / GP-BO / CMA-ES

**Budget** | `≤1000` total evals/config (≈973 GP + startup/jitter) · `ConfigTrialBudget` cap kills the `×27` outer/remediation multiply

**Objective** | DD-util booster (hard-wired): `3·ret_frac + ½·dsr - deficits - 5·dd_over - $10k base - 20/25 DD`

**CV** | CPCV C(5,2) · purge + embargo

**Ledger** | per-config trial counts emitted

### TECH STACK

`XGBoost - direction` | `FinBERT - sentiment gate` | `market-regime gate` | `conformal P(win)` | `hai_trainer.py` · RE-XX feats

### ASSET LIBRARY

- optimizer backend `71004f8`
- `trial_ledger_manifest.schema.json` (contract v1)
- parquet ledger (pyarrow)

### GUARDRAILS

- Full population — no top-K / "high-impact only" / freezing
- Every calibratable param searched (hardcode guard + hook)
- Converge within `≤1000` total evals · cap is per-version (never cross-deployed)
- Booster hard-wired ON — never env-gated off (return-within-budget)

## 3 - HAI TANDEM (R8)

3-MODEL OVERLAY · PER-CONFIG · TRAIN=SERVE (D15)

### THE NARRATIVE - SOUL

"Calibrate the system you actually trade." The HAI gate is a 3-model overlay — (1) XGBoost direction · (2) FinBERT sentiment · (3) market-regime — trained/wired in-cycle; the objective scores the full HAI-gated system. A neutral-stubbed (ML-only) gate is train=serve — forbidden (D15).

### FRAMEWORK - TANDEM

**PHASE** | **WHAT**

**Capture** | candidate trades + features (bootstrap)

**Train** | per-config HAI model on those labels

**Gate** | re-run HAI-ON · objective scores HAI-gated

**Pin** | cycle-pin `RX##+SHA` · train=serve

### TECH STACK

`XGBoost - direction` | `FinBERT - sentiment gate` | `market-regime gate` | `conformal P(win)` | `hai_trainer.py` · RE-XX feats

### ASSET LIBRARY

- `hai_models<<cfg><RX##><SHA>.pkl`
- `i_mlMnscore` · base-anchored (in population)
- RE-01.RE-12 (HAR-YZ · path sig · NetMm - OFI)

### GUARDRAILS

- Health gate BLOCKING: adv AUC<0.85 · per-dir AUC<0.55
- Live only with a health-passing, in-sync, cycle-pinned model
- HAI hyperparams searched per-config (R1/R3)
- Only the cycle-pinned per-config model gates live (train=serve)
- **D15:** all 3 gates (ML + sentiment + regime) on real inputs + train=serve, or signed-OFF — never neutral-stubbed

## 4 - ROBUSTNESS

WALK-FORWARD · CPCV · HEALTH

### NARRATIVE - SOUL

"Out-of-sample or it didn't happen." Purge-and-embargo walk-forward / CPCV on the HAI-gated system, plus the blocking HAI health gate.

Enforces **DR-1 robustness · R8 health**.

### FRAMEWORK

- S5 - WF / CPCV · purge + embargo
- G4 - purge-probe PASS
- G\_HAI - adversarial AUC<0.85 · per-dir AUC<0.55
- OOS path Sharpe on HAI-gated result

### TECH STACK

`walk_forward.py` | `cpvc.py` | `hai_adversarial.py` | `stress_runner`

### ASSET LIBRARY

- `conformal_calibration.pkl`
- `adversarial_auc.json` · per-direction report
- CPCV posterior + OOS slices

### GUARDRAILS

- Leak-free by construction · purge + embargo enforced
- HAI health gate is blocking
- Robustness binding to the charter gate

## 5 - HONEST CHARTER GATE

DR-1 · FULL 5-CRITERIA

### NARRATIVE - SOUL

"The real bar." The full 5-criteria charter gate with binding robustness, FIXED constraints, DSR-deflated — scoring the HAI-gated system (DR-1).

Enforces **DR-1 · Principle 3**.

### FRAMEWORK

- S7 - full charter (no advisory-only)
- DSR / PBO deflation
- G5 - no null required field ⇒ BLOK
- G8 - recompute within tolerance + HAI-Sync sign-off

### TECH STACK

`precision_gate.py` | `charter gate 5c20f21` | `auto_remediation`

### ASSET LIBRARY

- `precision_metrics.json` (with CIs)
- `decision.json` · SHIP / BLOK
- charter gate commit `5c20f21`

### GUARDRAILS

- Honest 5-criteria pass with binding robustness
- Constraints FIXED — identical across runs
- Scores the HAI-gated system only (R8)

## 6 - PACKAGING & SIGN-OFF

S8-S10 · INDEPENDENT RE-DERIVE

### NARRATIVE - SOUL

"Trust, but independently verify." Compliance Heat Map, packaging with the model SHA in the dossier, then an independent Validator re-derivation + HAI-Sync sign-off before READY.

Enforces **\$4.3 independence · R8**.

### FRAMEWORK

- S8 - Heat Map · six-field all-true (G6)
- S9 - packaging · params.json · model SHA · ALGO\_READY (G7)
- S10 - Validator re-derive on XOR-seeded holdout
- G8 - recompute within tolerance + HAI-Sync sign-off

### TECH STACK

`health_heatmap.py` | `archivist.py` | `validator (XOR_MASK)` | `HAI-Sync Validator`

### ASSET LIBRARY

- `validator_sign_off.json`
- HAI-sync sign-off (model SHA · train=serve · AUCs)
- `reproducer.sh` · full reproducible bundle

### GUARDRAILS

- Validator key ≠ Trainer key (\$4.3.1)
- No HAI-sync sign-off ⇒ no `READY_FOR_CANNON_API`
- Divergence ⇒ sign-off INVALID

## 7 - GOVERNANCE & OVERSIGHT

FLOW OVERSEES · LANES SELF-GOVERN

### NARRATIVE - SOUL

"Oversee, don't manage." Flow ratifies, adjudicates, and sets swimlanes per algo; it does not gate or hand-run lanes. Binding rules are mechanized as fail-closed gates, not restated verbally.

Enforces **Charter \$4 · \$9**.

### FRAMEWORK - AGENTS

- Calibrator · Trainer · Validator (independent)
- **HAI-Sync Validator** — R8 warden, gates promotion
- Sentinel · Researcher · Auditor · Adjudicator
- Per-lane git worktrees · commit hygiene

### TECH STACK

`7 per-agent keys` | `hai_sync_validator.md` | `OneDrive handoff feed` | `Obsidian mirror`

### ASSET LIBRARY

- `q7_calibration_coverage_rules` (R1-R8)
- `AMENDMENT_HAI_JOIN_CALIBRATION · RULING_FLOW_HAI_SYNC`
- Golden-source registry · swimlane board
- `FLOW_APPROVED_PLAN` (Calibration · S2/S6 · S7)

### GUARDRAILS

- Flow verifies / gates / escalates — does not edit producer/HAI math
- Validator independence (Charter \$4.3)
- Violating any of R1-R8 invalidates the cycle

## 7 - AGENT OPERATING MODEL (OVERLAY · OWN KEYS)

### Calibrator

`q7-calibrator` · nightly/strategy · Lanes 1-2 population + concerted search

### Trainer

`q7-trainer` · pre-cycle freshness · Lane 3 HAI tandem (per-config) search

### Validator

`q7-validator` · AFTER Trainer (\$4.3) · Lanes 4-6 robustness → charter gate → \$20 re-derive

### Sentinel

`q7-auditor` · continuous 10-min · monitors all lanes + Channel-1 telemetry

### Researcher

`q7-researcher` · weekly Sun · feeds population/features into Lane 1

### Auditor

`q7-auditor` · monthly · structural audit D1-D14 (Lane 6) · 3rd-line assurance

### Adjudicator

`q7-adjudicator` · event-driven · escalations / decision records (Lane 7)

### Flow

oversight · ratify / adjudicate / set swimlanes · Lane 7 · does NOT run lanes

CALIBRATION & VALIDATION METHODS

PRE-CYCLE - STAGE 0

Structural-Integrity Cornerstones — agent-led S-check gate

Proves Pine + Python + pipeline defect-free before trial #1 (CI-CB + a runtime ship-smoke). A clean model converges within ~1000 trials, so this kills the cause of the >1000-trial burn. Fall-closed, agent-signed cert gates the cycle.

SIGNIFICANCE - LANE 4

Monte Carlo — permutation test

Builds a null distribution by permuting labels/returns; the permutation p-value tests whether the edge is statistically real vs. random chance.

HAI GATE 3 - LANE 3

Market-regime — regime gate

Classifies bull/bear + vol regime; trades only in the favorable regime and scales exposure. AND-ed per-bar, thresholds calibrated, train=serve (D15).

OBJECTIVE - LANE 2

DD-utilization booster — return-within-budget

Hard-wired composite (3-ret\_frac + 1/2-dsr — deficits — S-dd\_over, \$10k base, 20/25 DD) replaces the DSR-ratio objective that collapsed position size — makes a strategy USE its drawdown budget. Never env-gated off.

STRESS - CHARTER GATE

Monte Carlo — stress simulation

Synthetic shock scenarios (vol shock · drift shock · fat-tail · bimodal) stress each config; it must survive to ship — the ship-gate's 4-scenario pass.

CALIBRATION - LANE 3

Conformal prediction — P(win)

Wraps the HAI score into a calibrated win-probability with coverage guarantees, so the gate threshold means what it says.

SEARCH - LANE 2

Bayesian Optimization — Optuna (TPE · GP-BO · CMA-ES)

Proposes each next trial from a probabilistic model of past trials — searches the full high-dim calibratable population (converging within a <1000 total-eval cap) for the objective-maximizing settings, far more sample-efficient than grid/random over hundreds of dims.

HAI GATE 1 - LANE 3

XGBoost — directional ML (per-config)

Gradient-boosted trees trained in-pipeline each cycle; predicts trade direction. Threshold resolved from OOS, never hardcoded (D14). One of three HAI gates.

OVERFIT CONTROL - LANE 5

DSR / PBD — Deflated Sharpe · Prob. of Backtest Overfitting

Deflates the Sharpe for the number of trials tested and estimates overfitting probability — rejects configs that only look good because many were tried.

OOS ROBUSTNESS - LANES 2-4

OPCV — Combinatorial Purged Cross-Validation C(5,2)

Trains/tests over many purged + embargoed path combinations so the score is leak-free out-of-sample — not one lucky split. Produces the lower-CI estimate used by the gate.

HAI GATE 2 - LANE 3

FinBERT — financial-sentiment gate

Pretrained transformer scores news sentiment per asset class; blocks trades into adverse sentiment. Needs a real feed + calibrated threshold + train=serve (D15) — never neutral-stubbed.

TEMPORAL - LANE 4

Walk-forward analysis

Rolling out-of-sample re-fit/test confirms the edge is stable over time, not specific to one regime.

16/16

HONEST SHIP TARGET PER CYCLE

100 %

CALIBRATABLE COVERAGE (R1)

≤ 1000

TOTAL EVALS / CONFIG · CONVERGE-WITHIN CAP (R7)

< 0.85

ADVERSARIAL AUC · HAI HEALTH

≥ 0.55

PER-DIRECTION AUC · HAI HEALTH

SHA

TRAIN = SERVE · CYCLE-PINNED (R8)

# ROLES & RESPONSIBILITIES

7-AGENT OPERATING MODEL · 3 LINES OF DEFENSE OVERLAY · RACI

## 1st line — owns the risk / runs the work

Strategy & Research (10), Calibration & Production (20), Platform & Technology (30), Growth (50). Calibrator / Trainer / Researcher / DevOps.

## 2nd line — independent challenge / oversight

Governance & Risk (00) = Flow; Compliance & Legal (60). Sets gates & limits, ratifies, adjudicates — does not run lanes.

## 3rd line — independent assurance

Audit & Assurance (70) = Auditor / Adjudicator. Never produced by the 1st line that made the artifact (enforced by per-agent keys).

## KEY ISOLATION (CHARTER §4.3)

Each agent runs under its OWN Anthropic key; **Validator key ≠ Trainer key**. The Anthropic billing dashboard is the external integrity check. The control plane asserts isolation before every judgment node and fails closed if a key is shared.

## HONEST GATE

Nothing ships on a *claimed* GREEN — promotion gates on the cert artifact (GREEN + unlocked + sha == R-PROMOTE). Verifier ≠ producer.

## 7-AGENT OPERATING MODEL

AGENT	KEY	TRIGGER	OWNS (LANE)
Calibrator	q7-calibrator	nightly / strategy	population + concerted search
Trainer	q7-trainer	freshness pre-cycle	per-config HAI tandem
Validator	q7-validator	after Trainer (§4.3)	robustness - charter gate - §2D re-derive
Sentinel	q7-sentinel	continuous 10-min	monitoring - telemetry
Researcher	q7-researcher	weekly (Sun)	alpha / feature research
Auditor	q7-auditor	monthly (1st)	structural audit D1-D14
Adjudicator	q7-adjudicator	event-driven	escalations - decisions

## RACI (BY FUNCTION)

FUNCTION	RESPONSIBLE (1ST)	ACCOUNTABLE	CONSULTED (2ND)	ASSURED (3RD)
10/20 Calibration & Research	Calibrator / Trainer / Researcher	Founder	Validator / Flow	Auditor
30 Platform	DevOps / UI Platform	Founder	Flow	Auditor
40 Capital & Finance	CFO	Founder	Compliance / Counsel	Auditor
50 Growth	Head of Mktg/UX	Founder	Compliance	Auditor
60 Compliance & Legal	Compliance	Founder	Counsel	Auditor
00 Governance	Flow	Founder	—	Adjudicator

Source: 00\_Governance\_InformationArchitectureStandard §5 RACI + Charter §4.1 seven-agent model. Founder = Accountable across functions.