TARGET OPERATING MODEL

Mother blueprint · the second stage of the design chain.

Vol: 18 | v0.6.3



49G Ltd.
ADGM, Abu Dhabi, UAE

Target Operating Model

Mother blueprint · the second stage of the design chain.

Release: Library · version v0.6.3, pinned by commit; no tag · status DRAFT, fixed test NOT_RUN: not yet run
Source: 49g-holding/49GAION-target-operating-model · blueprint/TOM-BLUEPRINT.md @ 3b0f5e3
Precedence: 4 · Library part, DRAFT, NOT_RUN
Built: 31 October 2026 · by 49AIOC-OCO, from the Markdown at that commit; no word changed

© 2026 49G Holding Ltd (Abu Dhabi Global Market), the owner of 49GAION · texts licensed CC BY 4.0 · the Markdown on GitHub is the source and governs
Powered by 49GAION

```
# ── Blueprint metadata (machine-readable; the panel generator and the back-office
builder read this block) ──

blueprint:

  aioc: "[ORG]AIOC-TOM"

  parent: "[ORG]AIOC-CH"                  # example code of the Charter; each organ
isation writes its own Charter's code (Standard §1.7, item 13)

  full_name: "Target Operating Model"

  organisation: "<organisation>"          # filled in each organisation's own copy

  type: "specialised"

  org_aion: "[ORG]AION"

  blueprint_version: "v0.6.3"

  status: "draft"

  template:

    name: "49GAION AIOC Blueprint Template"

    version: "v1.5.0"

    source: "github.com/49g-holding/49GAION-aioc, template/AIOC-BLUEPRINT-TEMPLATE.
md @ 06b00d9"

  mother:

    name: "49GAION Target Operating Model"

    version: "part card v1 (DRAFT)"

    source: "github.com/49g-holding/49GAION-protocol, library/cards/target-operatin
g-model.part-card.json and the Protocol section Target Operating Model (§19); Stand
ard §3.1, item 4 (49GAION DR-0016). This text is its mother blueprint"

  writing_repository: "<org>aioc-tom"

  code_project: "<org>aioc-tom, on the organisation's code platform"

  panel_register_id: "none yet"

  capabilities:

    standard: ["Chief of Staff", "Office", "Input socket", "Panel"]

    functional: ["Request Reconstruction", "Needs Requests", "Structuring", "Entity
 Packaging", "Entity Model Design", "Hand-offs", "Operational Walkthrough", "Indepe
ndent Review", "Decision Package and Release"]
```

Blueprint · [ORG]AIOC-TOM · Target Operating Model · v0.6.3

Draft, closed at v0.4.0, 15 October 2026; v0.4.1 the same day changes wording only (decisions/DR-0004): every human position is a HOAC position, and no natural person exists in the system. v0.5.0, 16 October 2026, links this AIOC directly to the stages after it (decisions/DR-0005): a release notice to Process and Position Design on every filing, and the returns of Process and Position Design and of Delivery and Implementation received directly, each carried to an end under the Library's design return contract (§5.10). v0.6.0, 16 October 2026, adds what the model also states and reads (decisions/DR-0006; Annex H): the strategy on one page, how each capability is delivered and how mature it is, the critical value streams, each outside party's journey, the value case, the priority of each element, and records of the work as it runs today, never about a person. v0.6.1, 19 October 2026: the Chief of Staff takes its COS code [ORG]CELL-COSTOM (decisions/DR-0007). Written by [ORG]AIOC-OCO on the directions recorded in decisions/DR-0001, decisions/DR-0002, decisions/DR-0003, decisions/DR-0005 and decisions/DR-0006. Version 0.4.0 adds four features to the closed v0.3.0, and removes or weakens nothing of it: a register of the roles across the release (§5.6; Annex E.17), a run depth (Annex F), two independent authors at Frame and Design (Annex G), and reference models from the 49GAION Library (§5.5; Annex B.5). Every rule here either rests on a 49GAION text, is carried from the case study (docs/DESIGN-BASIS.md; full rule texts in docs/CARRIED-RULES.md), or is new in this AIOC and marked **New**. Nothing here has run. Every fixed test reads NOT_RUN.

TOM in one page

TOM (Target Operating Model) turns an organisation's own documents into a written model of how it should work. For each part of the organisation, the model sets out what the part must be able to do, how its main work flows, who decides what, and where it hands work to other parts and to outside parties. It names roles and what each decides. It does not say what fills a role.

It serves one person's work, a unit, a company, a group of companies, or a group with independent institutions inside it. Where there are several companies, it designs only the hand-offs between them and with outside parties.

What goes in

1. An approved design request, recorded in Source Lifecycle, the organisation's library of sources: who asks, what the design is for, what should be different, and how fine the model should be (light, standard or deep).
2. The organisation's own documents, read only by reference. The input socket checks them, cuts them into numbered passages, labels each passage with how far it can be trusted, and hands the engine one sealed Input Pack.
3. Answers to what the documents do not say. TOM never asks a person. It asks the Chief of Staff of Source Lifecycle, who gets the answer from the position that can give it, has it approved and recorded, and returns it. Each answer arrives as a new version of the Input Pack.
4. Where the organisation has them, records of how the work runs today, exported by role and by process and never about a person. Where there are none, the model says so and goes on.

Today Source Lifecycle is the only source. Other kinds of source plug into the same socket, each through its own adapter file, and only to be read.

What the engine does, in nine stages

1. Admit: check the request and the connection, and make the Input Pack.
2. Frame: work out what is asked, who may decide it, what exists today, what should change, and what is unknown or in conflict.
3. Structure: split the work into parts, and propose which to design first.
4. Child briefs: for each part, what it must achieve and how anyone would know.
5. Package: gather each part's own evidence, and nothing about the other parts.
6. Design: design each part's model, and write down why. A general pattern may suggest a structure, but it never counts as evidence.
7. Hand-offs: pair every hand-off between parts and with outside parties, list every role once with what it decides, and flag legal questions for a lawyer.

8. Walk: test the model against realistic situations, and record where it holds and where it stops.
9. Decide and release: the organisation decides each element; the decided model is frozen and filed back in Source Lifecycle, and Process and Position Design is told at once.

Each stage has one input, one output and one test that says when it is done. The frame and each part's design are written twice, separately, by two AI engines of different families; their differences are settled on the sources or put to the organisation. Inside every stage, three or more other independent AI engines check the output: each alone first, then in a short debate held to quotes. They do not vote. A human position, the TOM Design Lead, rules on what they still dispute.

What comes out: the TOM Release

1. The Decision Package: a plain summary, what is still unknown, and the choices to make.
2. For each part: what it must be able to do, how each capability is delivered and how mature it is, its main flows and which are critical, who decides what, and where it hands work over, each with its reasons and sources.
3. The hand-offs between parts and with outside parties, each outside party's journey through the organisation, and the legal questions for a lawyer.

3a. The strategy on one page, the value the change is meant to bring, the priority of each element, and how the work runs today where records exist.

4. The role register: every role in the model, what it decides, what it answers for, and what it hands to whom.
5. The test against situations, and the quality checks made at each stage.
6. What is still open: unknowns, assumptions, deferred items, and any defect the organisation chose to accept.
7. The record: the sources used, and the trace from each statement back to its source.

Who uses it

- The decision-maker who asked for the design decides each element: accept, correct or reject. Silence decides nothing.

- The positions that lead each part check its model against how the work runs.
- Process and Position Design, the next stage, is told of each decided release directly and reads it, starting with the role register. It sends back what it cannot use, and so may Delivery and Implementation, the stage after it.
- Every return goes back and forth until it ends: answered by a new release, answered by a decision, or set aside by decision and carried forward as a known defect or a decided assumption. None is left open.
- Auditors and lawyers use the sources, the trace and the record.

Closure. This blueprint is closed at v0.4.0; v0.5.0 adds its links to the stages after it, and v0.6.0 what the model also states and reads (Annex H); neither removes anything. Every stage is fixed by its input, its output and its done-test (§5.0), and every object a stage makes has its field list (Annex E; docs/CARRIED-RULES.md). It holds no open design question. What remains is to build and test it, and the decisions that only a human seat can take, listed once in §13.



Section 00

Identity and lineage

Target Operating Model

Field	Value
Code	[ORG]AIOC-TOM. Collision check (Naming Standard §5 and R6): the initials do not recreate AIOC, AION or 49GAION, and do not read as a version suffix. Each organisation records the code in its register before the repository is created (Naming Standard R9)
Full name	Target Operating Model
Description	"[ORG]AIOC-TOM is <organisation>'s Target Operating Model core, built from 49GAION Target Operating Model"
Kind	AIOC
Parent	<code of the Charter of this [ORG]AION>; the cards use the example code [ORG]AIOC-CH. A top-level AIOC: only the Charter has parent none (Standard §1.7, item 13). A change of parent is the listed type change-parent
Inner units	[ORG]CELL-TDL, [ORG]CELL-COSTOM, [ORG]POD-TRR, [ORG]POD-TS, [ORG]POD-TEP, [ORG]POD-TEM, [ORG]POD-TH, [ORG]POD-TW, [ORG]POD-TR: seven Pods with their seventeen Cells, and two Cells directly under the AIOC (§17). No inner AIOC
HOAC label	[ORG]HOAC-ACS, held by its top Cell, the TOM Design Lead [ORG]CELL-TDL (label_holder)
Accountable to	<code of the Cell carrying [ORG]HOAC-TM>; the cards use the example code [ORG]CELL-TMS (Naming Standard R15). That Cell signs this AIOC's install, its change-parent and its restart-stopped-unit (Standard §5.3, item 4; §4.4, item 3)
Type	Specialised: the organisation needs a decided target operating model (§16)
[ORG]AION	[ORG]AION, based on 49GAION Standard v1.5.0
Versioned Definition	this file; the Panel and the Back office are built from it
Organisation	<organisation>

Field	Value
Made from	the Library part 49GAION Target Operating Model, part card v1 (DRAFT), and the Protocol section of the same name (§19), merged on 15 October 2026 (49GAION/DR-0016)
Template	49GAION AIOC Blueprint Template v1.5.0 at 06b00d9
Mother	49GAION Target Operating Model, part card v1. This text is its mother blueprint (49GAION/DR-0016, item 7)
Blueprint version and status	v0.6.3 · draft, closed at v0.4.0 (the Closure statement above §0); v0.4.1 changed wording only (decisions/DR-0004); v0.5.0 adds the direct links to the stages after it (decisions/DR-0005); v0.6.0 adds Annex H (decisions/DR-0006); v0.6.1 gives the Chief of Staff its COS code [ORG]CELL-COSTOM (decisions/DR-0007)
Writing repository	<org>aioc-tom, plus -records for an organisation's runs and -records-restricted for run records that touch Strictly confidential sources or personal data (Naming Standard R9)
Code project	<org>aioc-tom, on the organisation's code platform
Panel	not yet built
Back office	not yet built
Attribution	Powered by 49GAION

Lineage. This AIOC joins six capabilities of an earlier organisation-design chain into one unit: request reconstruction, commission decomposition, entity work packaging, single-entity operating-model design, cross-entity assembly (mandated there but never designed; here narrowed to Hand-offs), and the operational walkthrough. A seventh, requester stewardship, is kept, split: its person-facing half is done by the Chief of Staff of [ORG]AIOC-SL; its other half (capturing the request, tracking what is missing, carrying decision rows) is done here (§5.1, §5.2, §5.9). docs/DESIGN-BASIS.md says, for every rule, why it is here and which test case checks it. docs/CARRIED-RULES.md holds the full text of the carried rules that this blueprint states only in short. The organisation's private register of the earlier texts is kept in its code project.

The earlier work is a case study. This AIOC was designed by studying an earlier organisation-design chain. That chain is a case study. Its rules are reused here, in full text in docs/CARRIED-RULES.md. Its tests and its run results count for nothing here. Its code is not wired in: every code step of this AIOC is written new against this blueprint, with its own tests, and the earlier code is kept unchanged in the code project only to be read. No earlier result is the reason for a rule, and nothing is left out or narrowed because of one. Every fixed test of this AIOC reads NOT_RUN until it runs.



Section 01

Card (49GAION Operating Protocol §3)

```
card:
  unit_kind: "AIOC"
  code: "[ORG]AIOC-TOM"
  parent: "[ORG]AIOC-CH"                  # example code of the Charter, the root A
IOC; only it has parent none (Standard §1.7, item 13)
  inner_units: ["[ORG]CELL-TDL", "[ORG]CELL-COSTOM", "[ORG]POD-TRR", "[ORG]POD-TS",
 "[ORG]POD-TEP", "[ORG]POD-TEM", "[ORG]POD-TH", "[ORG]POD-TW", "[ORG]POD-TR"]
  type: "specialised"
  org_aion: "[ORG]AION"
  card_version: "v0.6.3"
  input_door: "Chief of Staff"
  wake: { method: "queue", verified: false }
  receives:
    - kind: "run request"                 # a Work Item from the Requesting Authori
ty's Cell, through the HU input port; it carries only the reference of an approved
Design Request source in [ORG]AIOC-SL
      format: "Work Item (Protocol §4.1) whose sources field holds one Design Reque
st reference"
      acceptance_test: "TEST-TOM v5, cases 3, 29, 44, 81, 89"
      decision_needed: false
    - kind: "source reference"            # served through an adapter of the Input
socket; in this version, the output socket of [ORG]AIOC-SL
      format: "source reference (contracts/source-reference.schema.json)"
      acceptance_test: "TEST-TOM v5, cases 1-6, 31-34, 44-52, 119"
      decision_needed: false
    - kind: "needs answer"                # the output of a needs request sub-item
      format: "needs answer (contracts/needs-answer.schema.json)"
      acceptance_test: "TEST-TOM v5, cases 10-12, 35, 36, 49, 118"
      decision_needed: false
    - kind: "needs status"                # accepted, returned or asked, from [ORG]
AIOC-SL
      format: "needs status (contracts/needs-status.schema.json)"
      acceptance_test: "TEST-TOM v5, case 10"
      decision_needed: false
```

```
    - kind: "return"                      # a design return from [ORG]AIOC-PPD or [
ORG]AIOC-DI on a release of this AIOC (Protocol §19, items 5 and 6); carried to an
end by §5.10
      format: "design return (49GAION-protocol library/contracts/design-return.sche
ma.json, at the version pinned in contracts/PINS.json)"
      acceptance_test: "TEST-TOM v5, cases 101-107"
      decision_needed: true
  delivers:
    - kind: "needs request"
      format: "needs request (contracts/needs-request.schema.json), as a sub-item o
f the run Work Item"
      usual_receivers: ["[ORG]AIOC-SL"]
    - kind: "missing report"
      format: "Missing Report (contracts/missing-report.schema.json), one per pass"
      usual_receivers: ["[ORG]AIOC-SL"]
    - kind: "needs status"                # withdrawn or superseded, from this AIOC
      format: "needs status (contracts/needs-status.schema.json)"
      usual_receivers: ["[ORG]AIOC-SL"]
    - kind: "TOM Release filing"          # the release itself lives only in [ORG]A
IOC-SL; every later stage reads it there, by reference
      format: "TOM Release manifest (contracts/tom-release.schema.json) and its par
ts, filed as a source of origin kind operating model; filing receipt (contracts/tom
-filing-receipt.schema.json)"
      usual_receivers: ["[ORG]AIOC-SL"]
    - kind: "release notice"              # on every filing of a decided release (P
rotocol §19, item 5; §5.9 step 8a)
      format: "release notice (contracts/release-notice.schema.json): the release I
D, version and manifest hash, its filed reference, and the returns it answers; neve
r release text"
      usual_receivers: ["[ORG]AIOC-PPD"]
    - kind: "return answer"               # the end of a return, or its next round,
 recorded on the return itself (§5.10)
      format: "the return at its next version (49GAION-protocol library/contracts/d
esign-return.schema.json): its status, the answering release or decision reference,
 and for a return set aside, its carried_forward_on"
      usual_receivers: ["[ORG]AIOC-PPD", "[ORG]AIOC-DI"]
  anatomy:
```

```
    panel: { health: { tests: true, metrics: ["stuck", "returned", "late", "needs o
pen", "needs past deadline", "run state", "pack version", "reviews held", "review b
udget used", "run depth", "authoring", "role findings open", "returns open", "retur
ns past deadline", "return age", "return round", "strategy elements not given", "ac
tivity profile"] } }

    back_office: { converses_with_humans: false, hosts: ["Chief of Staff", "Office"
], protocol_traffic: true }

    definition: { location: "<org>aioc-tom, blueprint/TOM-BLUEPRINT.md", version: "
v0.6.3" }

    delivery: { rollback: "one-click" }

    connectors:

      - { role: "source of record", connector: "[ORG]AIOC-SL output socket", access
_managed_in: "panel" }   # the Panel shows and requests it; the grant itself is sig
ned at the tool gateway as connect-external-port by the [ORG]HOAC-ACS seat over [OR
G]AIOC-SL (Protocol §13, item 5; upstream/PROPOSALS.md, item 4)

      - { role: "next design stage", connector: "[ORG]AIOC-PPD input port: release
notices out, returns in", access_managed_in: "panel" }   # signed at the tool gatew
ay as connect-external-port by the [ORG]HOAC-ACS position over this AIOC (Protocol
§19, item 7; H-35)

      - { role: "returns from the last stage", connector: "[ORG]AIOC-DI output port
: returns in", access_managed_in: "panel" }   # signed as above (Protocol §20, item
 3; H-35)

    gateway: { metering_id: "<per-AIOC metering id>" }

  full_name: "Target Operating Model"

  version: "0.6.3"

  lineage: { id: "[ORG]AIOC-TOM", version: "0.6.2", relation: "original" }

  lifecycle_state: "designed"

  blueprint_version: "v0.6.3"

  template_version: "v1.5.0"

  accountable_to: "[ORG]CELL-TMS"         # example code of the Cell carrying [ORG]
HOAC-TM (Naming Standard R15); never a Cell inside this AIOC

  top_cell: "[ORG]CELL-TDL"

  hoac: "[ORG]HOAC-ACS"

  label_holder: "[ORG]CELL-TDL"

  fixed_test: { id: "TEST-TOM", version: "v5", owner: "[ORG]CELL-TMS" }

  skills:

    - { id: "tom-request-reconstruction", name: "Request reconstruction from source
s", tags: ["design", "intake"] }
```

```
    - { id: "tom-entity-model", name: "Entity operating model design", tags: ["desi
gn"] }

    - { id: "tom-hand-offs", name: "Hand-offs between parts and with outside partie
s", tags: ["design"] }

    - { id: "tom-walkthrough", name: "Operational walkthrough", tags: ["assurance"]
 }

    - { id: "tom-role-register", name: "Role register across the release", tags: ["
design"] }

    - { id: "tom-returns", name: "Returns from later stages, each carried to an end
", tags: ["coordination"] }

  ports:

    - { unit: "[ORG]AIOC-TOM", port: "ai-in", direction: "input", type: "AI", ident
ification: "technical-key", accepted_content: ["source reference", "needs answer",
"needs status", "return"], authority_limit: "read only, within the organisation-des
ign scope signed for this connection; returns only on the connections signed with [
ORG]AIOC-PPD and [ORG]AIOC-DI", audit_trail: "unit register and the serve log of [O
RG]AIOC-SL", required_tests: ["TEST-TOM v5"], enforcement: "shared-gateway-outside-
unit" }

    - { unit: "[ORG]AIOC-TOM", port: "ai-out", direction: "output", type: "AI", ide
ntification: "technical-key", accepted_content: ["needs request", "missing report",
 "needs status", "TOM Release filing", "release notice", "return answer"], authorit
y_limit: "requests, reports and filings only; files only sources of origin kind ope
rating model or design report, within the write scope signed for this connection; r
elease notices and return answers by reference only, on the connections signed with
 [ORG]AIOC-PPD and [ORG]AIOC-DI; never an instruction to a person", audit_trail: "W
ork Item record and the unit register", required_tests: ["TEST-TOM v5"], enforcemen
t: "shared-gateway-outside-unit" }

    - { unit: "[ORG]AIOC-TOM", port: "hu-in", direction: "input", type: "HU", ident
ification: "live-human-sign-in-at-gateway", accepted_content: ["run request from th
e Requesting Authority's Cell (a Design Request reference only)", "run control and
configuration from the TOM Design Lead (never a source)"], authority_limit: "the si
gner's authority_scope; nothing received here is a source", audit_trail: "Office re
cord", required_tests: ["TEST-TOM v5"], enforcement: "shared-gateway-outside-unit"
}

    - { unit: "[ORG]AIOC-TOM", port: "hu-out", direction: "output", type: "HU", ide
ntification: "live-human-sign-in-at-gateway", accepted_content: ["run brief to the
TOM Design Lead", "run preview to the TOM Design Lead", "ruling sheets to the TOM D
esign Lead", "release notice to the Requesting Authority's Cell"], authority_limit:
 "information only; never a question about the organisation", audit_trail: "Office
record", required_tests: ["TEST-TOM v5"], enforcement: "shared-gateway-outside-unit
" }

    - { unit: "[ORG]AIOC-TOM", port: "stop", direction: "input", type: "HU", identi
fication: "live-human-sign-in-at-gateway", accepted_content: ["stop"], authority_li
```

```
mit: "stop only", audit_trail: "unit register", required_tests: ["TEST-TOM v5"], en
forcement: "shared-gateway-outside-unit", stop_door: true }

  configuration:                          # every slot is set by the organisation (
§15); an empty slot fails the joining test

    office_accountable_human: { value: "<set by the organisation>", set_by: "<seat>
", date: "<date>", version: "<vN>" }

    central_head: { value: "<the OCO of this [ORG]AION>", set_by: "<seat>", date: "
<date>", version: "<vN>" }

    discussion_limits: { value: "<set by the organisation>", set_by: "<seat>", date
: "<date>", version: "<vN>" }

    waiting_deadlines: { value: "<per status>", set_by: "<Cell carrying [ORG]HOAC-T
M>", date: "<date>", version: "<vN>" }

    bounce_limit: { value: "<number>", set_by: "<seat>", date: "<date>", version: "
<vN>" }

    chain_depth: { value: "<number, at least 1>", set_by: "<seat>", date: "<date>",
 version: "<vN>" }

    halt_request_priority: { value: "immediate", set_by: "<seat>", date: "<date>",
version: "<vN>" }

    budget_cap: { value: "<amount per period>", set_by: "<Cell carrying [ORG]HOAC-T
M>", date: "<date>", version: "<vN>" }

    unattended_action: "read-record-triage"

    requesting_authority_cell: { value: "<set by the organisation>", set_by: "<Cell
 carrying [ORG]HOAC-GOV>", date: "<date>", version: "<vN>" }

    rows_per_pass: { value: 12, set_by: "<seat>", date: "<date>", version: "<vN>" }

    non_response_passes: { value: "<set by the organisation>", set_by: "<Cell carry
ing [ORG]HOAC-TM>", date: "<date>", version: "<vN>" }

    decision_passes_max: { value: 3, set_by: "<seat>", date: "<date>", version: "<v
N>" }

    explainers_per_pass: { value: 2, set_by: "<seat>", date: "<date>", version: "<v
N>" }

    round_caps: { value: "Request Reconstruction 12; Structuring 8; Entity Model De
sign 8", set_by: "<seat>", date: "<date>", version: "<vN>" }

    repair_cycles: { value: 2, set_by: "<seat>", date: "<date>", version: "<vN>" }

    situation_mix: { value: "standard library ≥ 60%, blind pack ≥ 30%", set_by: "<s
eat>", date: "<date>", version: "<vN>" }

    parallel_runs: { value: 1, set_by: "<seat>", date: "<date>", version: "<vN>" }

    review_engines: { value: 3, set_by: "<seat>", date: "<date>", version: "<vN>" }
```

```
    review_families_min: { value: 2, set_by: "<seat>", date: "<date>", version: "<v
N>" }

    reduced_review_permitted: { value: "no", set_by: "<Cell carrying [ORG]HOAC-TM>"
, date: "<date>", version: "<vN>" }

    review_budget_per_run: { value: "<set by the organisation>", set_by: "<Cell car
rying [ORG]HOAC-TM>", date: "<date>", version: "<vN>" }

    lazy_atomising_threshold: { value: 1000, set_by: "<seat>", date: "<date>", vers
ion: "<vN>" }

    lenses_max: { value: 6, set_by: "<seat>", date: "<date>", version: "<vN>" }

    run_depth_permitted: { value: "standard", set_by: "<Cell carrying [ORG]HOAC-TM>
", date: "<date>", version: "<vN>" }

    single_author_permitted: { value: "no", set_by: "<Cell carrying [ORG]HOAC-TM>",
 date: "<date>", version: "<vN>" }

    return_round_deadline: { value: "<set by the organisation>", set_by: "<Cell car
rying [ORG]HOAC-TM>", date: "<date>", version: "<vN>" }

    return_rounds_max: { value: "<set by the organisation>", set_by: "<Cell carryin
g [ORG]HOAC-TM>", date: "<date>", version: "<vN>" }
```

Joining the protocol. This AIOC joins only after every input and output on this card passes fixed samples (the joining test, Protocol §11, item 1; Standard §3.2, item 4). It also needs Source Lifecycle to declare, on its card, support for the contracts of docs/INTAKE-WITH-SOURCE-LIFECYCLE.md §9 (gate T0). It also needs the engine register at the tool gateway to hold eligible engines for the organisation's highest confidentiality level (T0, condition (g)), and two author families (condition (j)). Until then its lifecycle_state stays designed and no Work Item is routed to it. The cards of its Pods and Cells are set in §17 and generated into cards/.



Section 02

Organisation table and placement

2.1 Organisation table

Term in the blueprint	In <organisation>
The Top Office (decisions are signed by human Cells carrying HOAC labels)	<filled by the organisation>
[ORG]AIOC-OCO	<filled>
The Charter of this [ORG]AION	<code>
The Cell carrying [ORG]HOAC-TM	<code>
The engineering division	<filled>
The IT and infrastructure division	<filled>
The builder of every tool	<filled>
The Library blueprint this core is made from	49GAION Target Operating Model, part card v1; this text is its mother blueprint
The Requesting Authority: the seat that asks for the design and decides it	<code of the human Cell carrying [ORG]HOAC-GOV>, or of a human Cell carrying a HOAC label whose authority_scope, delegated by that Cell under delegate-authority, covers the design decisions of this AIOC (Standard §4.4, item 8). Its code is the answering_cell of every non-delegable row, except a row of a listed type, whose answering_cell is the Cell that signs that type (for set-purpose-or-values, the Cell carrying [ORG]HOAC-GOV)
The TOM Design Lead position	a HOAC position ([ORG]HOAC-ACS), a legal persona. Who holds it is the organisation's own mechanism, outside the system; this AIOC never sees it (Standard §1.1, item 9; 49GAION/DR-0049). The Design Lead position and the Requesting Authority's position are two positions, two parties: T0 fails when they are the same Cell (TEST-TOM, case 29). Whether one person holds both is the organisation's own mechanism, which this AIOC never reads

Term in the blueprint	In <organisation>
The source of record	[ORG]AIOC-SL: the only source this AIOC asks and files into. Sources of other types may be read beside it, each through its own adapter (§4.3).
The engine register	held at the tool gateway, which is carried by the Charter and run as code beside the core, outside the reach of every AI (Standard §4.4, item 11; 49GAION/DR-0052); read by this AIOC (Annex E.12)

2.2 Placement

- **Where it sits.** A top-level AIOC under the Charter, beside [ORG]AIOC-SL. It is the second stage of an organisation's design: the "next stage" of Protocol §13, item 6, set out in the Protocol section Target Operating Model (§19).
- **Its role.** It turns an organisation's own sources into a target operating model, decided by the organisation's own authority. It names roles and what each decides. It does not say what fills a role.
- **Scope.**
  - **It does:** reconstruct the request and write the Design Brief from sources; ask, through Source Lifecycle, for what the sources lack, and report what is missing; structure the work into children; brief each child; package each child in isolation; design each child's operating model with its rationale; record the hand-offs between children and with outside parties; list every accountable role-type once, in the role register; state the strategy frame from the sources and the organisation's decisions; record each capability's delivery model and maturity, each value stream's criticality and each outside party's journey; compile the value case, each element's proposed priority and the activity profile (Annex H); walk the model through realistic situations; have every stage's output checked under the Review Panel rule; prepare the decision; release the model.
  - **It does not:** ask any person anything about the organisation; write job descriptions; set workloads; assign work to anything other than an accountable role; choose tools or vendors; write processes, procedures or policies; write a success criterion; write a goal, a target, a level or a figure

of its own; watch work, or read a record that says who did a piece of work; make a legal determination; decide anything for the organisation.

- **Shapes it serves.** Any shape: a single person, a single entity, a group of entities, or a group with independent institutions inside it (the same four shapes as Source Lifecycle). A unit of an organisation may be the subject of a design (scope.subject, docs/OUTPUT.md §3).
- **Boundaries with other AIOCs.**
  - **[ORG]AIOC-SL** holds every source and every contact with a person. This AIOC holds only references and its own design records. It asks; the Chief of Staff of [ORG]AIOC-SL gets the answer and records it. Source Lifecycle never designs; this AIOC never records a source, except its own filed release and design reports.
  - **[ORG]AIOC-PPD**, the next stage, is told of each decided release directly (§5.9 step 8a), reads it from Source Lifecycle starting with its role register, and returns what it cannot use (§5.10). **[ORG]AIOC-DI**, the last stage, may also return what it cannot use. This AIOC writes no position or job description, and assigns work only to accountable roles.
  - **[ORG]AIOC-OCO** installs, fixes and upgrades this AIOC; it never grades it (Standard §2.3, item 9).
  - **Installation.** The core designs and proposes [ORG]AIOC-TOM, the governing seat approves it, and the Top Management Group installs it in the operation and uses its filed TOM Release (the complete definition, Part E). The AIOC itself belongs to the [ORG]AION. It is also a product on its own (docs/OUTPUT.md §14).
- **Overlap.** None known. Where an organisation already holds earlier design capabilities of the same kind, their content is carried here (docs/DESIGN-BASIS.md) and they are retired through OCO.



Section 03

Positions

Target Operating Model

Position (standard title)	Purpose	Reports to	Run today by	Cover
TOM Design Lead	the accountable seat of this AIOC and the owner of its path: owns the fixed tests of its parts; rules on contested review findings; accepts a reduced review level within the TM Cell's permission; decides a re-base; attests the method. Answers for the method, never for the organisation's choices	the Cell carrying [ORG]HOAC-TM	a HOAC position ([ORG]HOAC-ACS); who holds it is the organisation's own mechanism, outside the system, never seen by this AIOC	the organisation's own mechanism, outside the system
Chief of Staff	keep each run moving, honest and traced; the only door to Source Lifecycle	TOM Design Lead	an agent ([ORG]CELL-COSTOM)	none; the run halts
Source Proxy	answer only from the sources, with a citation, or refuse	Chief of Staff	an agent ([ORG]CELL-TSP)	none; the step halts
Request Analyst	reconstruct the request into the Design Brief; derive and read the lenses; write the party register; raise only the needs that pass the leverage test; author A of the frame	Chief of Staff	an agent ([ORG]CELL-TRA)	none; the step halts
Second Request Analyst	write the frame a second time, blind to the Request Analyst's draft, on an engine of the other author family (Annex G)	Chief of Staff	an agent ([ORG]CELL-TSRA)	none; the step halts
Frame Reconciler	reconcile the two locked frame drafts into one, under Annex G.5	Chief of Staff	an agent ([ORG]CELL-TFR)	none; the step halts

Target Operating Model

Position (standard title)	Purpose	Reports to	Run today by	Cover
Work Structurer	propose the children of the request	Chief of Staff	an agent ([ORG] CELL-TWS)	none; the step halts
Packager	decide each child's own evidence and declare every crossing	Chief of Staff	an agent ([ORG]CELL-TP)	none; the step halts
Designer	design each child's model and its rationale together, as author A	Chief of Staff	an agent ([ORG]CELL-TD)	none; the step halts
Second Designer	design each child a second time, blind to the Designer's ledger, on an engine of the other author family (Annex G)	Chief of Staff	an agent ([ORG] CELL-TSD)	none; the step halts
Ledger Reviser	reconcile the two locked design ledgers (Annex G), then revise the result in a fresh context	Chief of Staff	an agent ([ORG] CELL-TLR)	none; the step halts
Hand-off Analyst	record each clean pair of crossings, map every counterparty by class, list the perimeter, register the deferred classes	Chief of Staff	an agent ([ORG] CELL-THA)	none; the step halts
Situation Author	write walkthrough situations blind to the design	Chief of Staff	an agent ([ORG] CELL-TSA)	none; the step halts
Situation Walker	walk and classify under the binding rules	Chief of Staff	an agent ([ORG] CELL-TSW)	none; the step halts
Blind Reviewer, seats A–E	review a locked output, alone first, then in debate; one engine per seat per review	Chief of Staff	agents ([ORG]C ELL-TBRA to [O RG]CELL-TBRE)	none; any seat covers another

- The **Requesting Authority** is not a position of this AIOC (§2.1). It decides the design; this AIOC only prepares the decision.

- The Process Auditor of v0.2.0 is not a position. The process audit is kept as review point RP1b (§5.8).
- No AI Cell signs anything (Protocol §10, item 3). Each AI Cell's bot has its own identity card (Annex A.5 of the template; §17).
- **No natural person** (Standard §1.1, item 9, proposed by 49GAION/DR-0013). Every human position this AIOC deals with, its own and the organisation's, is a HOAC position, a legal persona. This AIOC addresses positions only, through their HU ports; who holds them, and how, is the organisation's own mechanism, outside the system, and no part of this AIOC holds, reads or infers it.
- Each position's role pack is carried in the role cards of the capabilities it performs (§4, §5).



Section 04

Standard capabilities

Defaults are in docs/STANDARD-CAPABILITIES.md of 49GAION-aioc v1.5.0. The differences are listed in §11.

4.1 Office

Definition

Default · form: **single Chief of Staff**. An Office forms only if the Design Lead needs more than one Chief of Staff.

Process (SOP)

Default.

Role card

None while the form is a single Chief of Staff.

Fixed test

TEST-TOM-COSTOM v5 (TEST-TOM v5, case 25: an Office is proposed, never self-created) · result NOT_RUN

4.2 Chief of Staff

Definition

Default, **managing** kind, with these additions:

- **The only door to Source Lifecycle.** It is the only Cell of this AIOC that sends anything outside it. It opens every needs request, Missing Report and filing as a Work Item to [ORG]AIOC-SL, and receives every answer through the Input socket. The connection's two external ports sit on its card (§17).
- **The door to the stages after it.** It sends each release notice to [ORG]AIOC-PPD (§5.9 step 8a), receives the returns of [ORG]AIOC-PPD and [ORG]AIOC-DI, and carries each to its end (§5.10). It never answers a return itself: the Requesting Authority does. Those ports sit on its card too (§17).

- **No question to a person.** It never asks any person anything about the organisation. Every question for the organisation goes to the Chief of Staff of [ORG]AIOC-SL as a needs request (§5.2). Its only human contact is its own HOAC seat, the TOM Design Lead, and its briefs to the Design Lead are about the run. Nothing the Design Lead sends is a source (§4.3).
- **An orchestrator that never judges itself.** It runs the gates of each run (Annex C): it dispatches the Cells, locks outputs, runs the code steps, draws the Review Panel and writes the run record. It records verification labels and review findings; it never writes a label on a check it ran, and never dismisses or downgrades a finding (§5.8; Annex B.4). It runs the Input socket's code steps (§4.3) and the review code steps (§5.8). It runs the compare code step (Annex G.4) and the role-register code step (§5.6 step 6a). It never rules on a finding, never dismisses or downgrades one, and never lowers a review.

Process (SOP)

Default, plus:

1. **Admission (stage S0).** On a run request: open the run record; have the Input socket build Input Pack P0 at the snapshot each adapter pins (§4.3); check T0 conditions (a) to (f) and (i) (Annex C.2) and read the engine register → P0 and the T0 record so far (evidence: the T0 record, with P0's pack_id).
2. **Run preview.** Send the TOM Design Lead the run preview (Annex E.13). Record the acknowledgement and, where only a reduced level is possible, the acceptance within the TM Cell's permission (§5.8 rule 18), and, where two authors at S5 would lower its review, the acceptance of single authoring there within the TM Cell's permission (Annex G.7); then check conditions (g) and (j) → acknowledged preview (evidence: the signed record).
3. **Pack review.** Run review point RP0 (§5.8). On its settlement, record T0 → T0 passed (evidence: the RP0 record).
4. **One run at a time.** One run per organisation (work in progress = 1). The slot is released only by the Requesting Authority's decision on the release, or by the run's closure (Annex C.1); never by a review verdict. A second run in parallel needs the TOM Design Lead's approval, recorded in the run record → run slot (evidence: run record).

5. **Passes.** Collect the needs requests the Pods raise; batch them into passes of at most the configured rows (§15, default 12), one decision per row; send each pass with its Missing Report (§5.2). Each pass is reviewed at RPa (§5.8) before it is sent → pass sent (evidence: Work Item IDs in the needs log).
6. **Waiting.** Overdue requests follow the [ORG]AION's waiting deadlines and escalation (Protocol §4.6, item 1): reminded, then raised to the TOM Design Lead and shown on the Panels of both AIOCs. A run is never left silent: past the configured passes it stops as non-response and emits a Readiness Report or the narrowest honest brief (§5.1, failure; Annex C.1) → ageing record (evidence: the needs log).
7. **Answers.** Hand each answer to the Input socket, which admits it, through the acceptance test of §4.3, into a new pack version. Re-open only the work that cites an affected atom → re-opened items (evidence: the pack's increment.reopens).
8. **Stops.** A stop takes effect at the tool gateway on every Pod and Cell of this AIOC; the Chief of Staff takes no action of its own and is itself stopped. After a signed restart-stopped-unit, it resumes from the records → nothing (evidence: the unit register).
9. **Revocation.** If the Source Lifecycle connection is revoked or narrowed, every later read is refused at the gateway. The run is held as held: connection revoked (Annex C.1). This is not a stop, and no restart is needed when a new connection is signed → held run (evidence: run record; unit register).
10. **Re-base.** Only on the TOM Design Lead's run-control decision: have the socket make a re-base pack version, with its diff; re-open only the work resting on changed atoms → re-base record (evidence: the diff and the decision).
11. **Returns.** Run §5.10 for every return received, whether or not a run is open → each return carried to its end (evidence: the returns log, Annex E.21).

Role card

Field	Content
Name	Chief of Staff
Position and seat	the position Chief of Staff; accountable seat: the TOM Design Lead ([ORG]HOAC-ACS)
Performed by	an agent ([ORG]CELL-COSTOM, full name TOM Chief of Staff)
Purpose	keep each run moving, honest and traced; be this AIOC's voice to the Design Lead and its only door to Source Lifecycle
Authority limits	alone: dispatch, lock, record, batch and send needs requests and Missing Reports, run code steps (the socket's and the review's included), file a release whose final status is decided, send its release notice, admit or refuse a return by the contract, generate its brief and send its rows and its answer; needs the TOM Design Lead: a second run in parallel, a budget above the cap, any change to configuration, a ruling on a review finding, a re-base, a reduced review level for the run, single authoring at S5, any review budget above the run's; never: ask a person anything about the organisation, answer a needs request itself, take anything from the Design Lead as a source, decide for the Requesting Authority, rule on or dismiss a finding, sign, close a Work Item this AIOC performed, lower the engines, families, rounds or level of a review, write a verification label, set or change the run depth, join two role names or settle a role finding, settle a difference between two drafts, run a two-author stage with one author outside Annex G.7, answer or classify a return, edit a filed release, let a return wait without its round and deadline
Write location	the run record and this AIOC's Work Items
Reports to	the TOM Design Lead
Escalation	to the Design Lead in one line, with evidence linked; a stop goes to the stop door, never through this Cell

Field	Content
Tools and access	the connection of each adapter (in this version, Source Lifecycle's), and the connections with [ORG]AIOC-PPD and [ORG]AIOC-DI, through the tool gateway, within their signed scopes; the code steps of the code project; no credential in any instruction
Deputy and cover	none; on failure the run halts and the Design Lead is told
Fixed test and score	TEST-TOM-COSTOM v5 (§9), plus the default Chief of Staff test (a)-(h)
Card version and status	v0.6.0 · draft

Fixed test

TEST-TOM-COSTOM v5 · result NOT_RUN

4.3 Input socket

Definition

Default, with this AIOC's standard format, the **source reference** (contracts/source-reference.schema.json), and these parts.

1. What the socket hands to the engine: the Input Pack. The engine reads nothing but the Input Pack of its run, at the version each step names. The Input Pack is one versioned, content-addressed object (contracts/input-pack.schema.json; Annex E.1). Its ID is TOMI- + the SHA-256 of its body. It holds references, hashes and labels only. The text it points to is resolved into the run workspace and never written into a record. It holds: the run; the Design Request, with the run depth it states; the adapters read, each with its grant, snapshot and test; the source catalogue; the atoms; what was refused, and why; the unknowns; the increment of this version; any carried release; the pack report. It never carries this AIOC's design content, anything received through a human port, a source's text, or a reference model (Annex B.5).

2. Pack versions: the rules.

1. P0 is made at gate T0, from the snapshot each adapter pins.
2. A new version is made only when:
   - a needs answer or a decision is admitted; it is read at the snapshot the answer names;
   - a notice changes the status, scope or classification of a reference already read;
   - the read plan opens a source not yet atomised (lazy atomising, slot lazy_atomising_threshold);
   - the TOM Design Lead decides a re-base.
3. A version keeps every earlier atom unchanged. It only adds atoms or flags them. Nothing is removed.
4. Each step names the pack version it reads, for the whole step. A newer version never changes a running step or review. Afterwards it re-opens only the work that cites an affected atom (increment.reopens).
5. **No silent re-pin.** Re-pinning the whole snapshot is never automatic. A **re-base** is a run-control decision of the TOM Design Lead. It is recorded with the diff between the old and new snapshots, makes a new version (made_because: re-base), and re-opens only the work resting on changed atoms.
6. **Supersession after the pin** does not stop the run. The affected atoms are flagged superseded since pin in the next version, and the Decision Package lists them in plain words. A revocation or a reclassification holds the run (held: connection revoked, or the hold of §7).
7. **The trace (T9).** Every element of a release traces to an atom of a pack version of this run, or of a pack version named in the manifest of a release this run carries forward. Each such atom still resolves, by hash, at its snapshot. The manifest lists every pack version with its snapshots (input_packs).

3. Labels, once. The socket cuts each held source into atoms and assigns each atom its Annex B label once, at admission, by Annex B.1 and B.1a and the

adapter's mapping. Every stage cites atoms of the pack. No stage creates an atom.

4. Adapters. Every source type plugs in through its own adapter file, adapters/<source-type>.adapter.md, with its fixtures and its own fixed test (Annex E.2). An adapter is the Standard's converter for one source type (STANDARD-CAPABILITIES §4.3). It only reads. This version has one adapter, ADP-SL, for Source Lifecycle. The adapter register is adapters/README.md. Three rules:

1. Adding an adapter is a tested change: one new file, its fixtures and its test, and a row in the adapter register (adapters/README.md). No stage changes.
2. An adapter is never a precedence rule. Two sources that disagree are a contradiction (Annex B.2), whichever adapters served them.
3. The same bytes served by two adapters are one source, with two references.

5. What Source Lifecycle serves, and what it is asked to add.

- **What Source Lifecycle serves today** (Protocol §13, item 4): the source ID, version and SHA-256 hash, and held or not held, within the approved scope.
- **What Source Lifecycle is asked to add** (docs/INTAKE-WITH-SOURCE-LIFECYCLE.md §9): the snapshot ID; the origin kind (document, explainer, confirmation, status, export, decision, operating model, design report); the status from Source Lifecycle's own list (draft, in review, approved, in force, superseded); the confidentiality level; for an export, the date it is as of; and, for an item derived from conversation, each statement's own Source Lifecycle label with its anchor (SL-A7). It is also asked to pin snapshots, read a needs answer at the snapshot that answer names, and keep every snapshot named by a filed release readable (SL-A8, as amended), and to send notices of revocation, narrowing, reclassification and supersession (SL-A11). **New in v0.6.0.** It is asked to admit an export of records of the work only as the organisation's projection by role-type and process, after its impact assessment and its HF-17 decision (SL-A13; Annex H.8), and to show and record the priority of an element row (the v0.6 sockets contract). Until Source Lifecycle serves these fields, gate T0 fails (Annex C). How the adapter reads each field, with the defaults where a value is not served, is fixed in adapters/source-lifecycle.adapter.md.

6. not held. A not held answer is recorded in the pack's unknowns, as an unknown with its cost. It is never taken to mean that the fact is absent.

7. What is refused.

- **New. Nothing that arrives through a human port is a source.** A run request is admitted only if it carries the reference of an approved Design Request in Source Lifecycle; the reference is admitted, never the words around it. Direction from the TOM Design Lead covers only method, configuration and run control (start, pause, stop, a second run, the budget, a re-base). A statement about the organisation in it is refused under case 2, and, if the design needs it, is asked through Source Lifecycle like any other need.
- **New.** A source that is not served by reference, or that comes from any channel other than an adapter's connection, is refused with its reason. Words typed or spoken to this AIOC are not a source.
- **New in v0.6.0.** An export of records of the work whose records still carry a field that says who did the work is refused person-level records, and nothing of it is kept (Annex H.8).
- The refusal reasons are those of Annex E.1 (refused[].reason).

8. The acceptance test of a needs answer (it makes case 11 concrete). An answer is admitted into a pack version only when all eight hold:

1. It validates against contracts/needs-answer.schema.json.
2. Its request_id and version match an open request.
3. Its outcome is allowed for the request's type.
4. Each source reference resolves, at the snapshot it names, to exactly one valid version with a matching hash.
5. For a non-delegable row, deciding_cell equals answering_cell.
6. For a listed type, signature_ref is valid for the exact row payload.
7. For a decision row, weak_approval_signal and delegated are both false.
8. It carries no secret.

9. A carried release enters P0 under carried (§5.9, Slices and carry-forward).

Process (SOP)

Default, plus the socket's five steps (STANDARD-CAPABILITIES §4.3), run by the Chief of Staff through the socket's code steps:

1. **Receive.** Three kinds of item: the run Work Item, through the HU port, carrying one Design Request reference only; references served by an adapter; needs answers and needs status messages.
2. **Identify.** For each item: the adapter, the grant, the time read, and the ID, version and hash.
3. **Normalise, through the adapter.** Make the canonical reference; map origin kind, status and confidentiality by the adapter's tables (defaults where unserved); cut each held source into atoms by the adapter's anchor scheme; assign each atom its Annex B label once, here, by Annex B.1 and the adapter's mapping.
4. **Check.** Each check passes, or refuses the item with its reason (refused[].reason):
   - the item resolves to exactly one valid version, and the hash matches (Protocol §8, item 2);
   - it is inside its scope, the scope is not expired, and the connection is not revoked;
   - it carries no secret;
   - an export of records of the work carries no field that says who did the work (Annex H.8);
   - instruction-like text is flagged, and kept as data;
   - for a Work Item: it is within the card;
   - for a needs answer: it passes the acceptance test of part 8 of the Definition, above.
5. **Deliver.** Write the next pack version and its pack report. Hand the list of re-opened work to the Chief of Staff.

Role card

Default; performed by the Chief of Staff ([ORG]CELL-COSTOM) through the socket's code steps and its adapters (Definition, part 4).

Fixed test

TEST-TOM-COSTOM v5 (TEST-TOM v5, cases 1-6, 29, 31-34, 44, 46, 49-52, 61, 81, 89, 96, 119 and 120); each adapter's own test (TEST-TOM-ADP-SL v2); plus the default Input socket test · result NOT_RUN

4.4 Panel

Definition

Default. The Panel of this AIOC also shows, for each run:

- the run state (Annex C.1), with its flags and their ages;
- the current pack version, its totals, and its difference from the previous version;
- per adapter: its last read, its scope expiry and the notices received;
- every needs request, with its state and its age against the waiting deadline;
- the latest Missing Report;
- each gate, with its result and evidence;
- per review: its point, level, verdict, findings by state, holds and budget used; any MATERIAL finding withdrawn by its raiser (§5.8 rule 14);
- the findings of the walkthrough, and each ruling;
- the run preview;
- the run depth and its source;
- per stage S1 and S5: two authors or one, and each job's differences by disposition;
- the role findings, by kind and state;
- the reference-model catalogue version, or unavailable;

- the strategy frame, with each element not given; the activity profile, or no observation (Annex H);
- the decision state of the release.

Across runs, it also shows every return received (§5.10): its sender, kind and elements, its round and the deadline of that round, its status and its age, and the decision rows open on it. A return past its deadline is shown in red, here and on its sender's Panel.

This is the Panel. It is not a Review Panel (§5.8).

Process (SOP)

Default.

Role card

Default.

Fixed test

Default Panel test · result NOT_RUN



Section 05

Functional capabilities

Nine stages (§5.0) are performed by the Input socket (§4.3: S0) and seven of these capabilities: Request Reconstruction (S1, S3); Structuring (S2); Entity Packaging (S4); Entity Model Design (S5); Hand-offs (S6); Operational Walkthrough (S7); Decision Package and Release (S8). Needs Requests (§5.2) is a service every stage uses. Independent Review (§5.8) is the rule that closes every stage. Each capability is written in the four-part form (Annex A of the template). Its Definition has one added row, **Done when**, which names its gate. Each step reads: **position**: action → output (evidence). Where a capability has more than three steps, its Steps cell points to a numbered list below the table. A step run by code is written as the Chief of Staff running a named code step: code is a tool, never a seat. Rules that apply to every capability are in Annex B (labels and the honesty ceiling), Annex C (run states, gates, Gate A and admissible criteria), Annex D (what this AIOC never does) and Annex E (the field lists of the new objects). Where a rule is carried from the case study, docs/DESIGN-BASIS.md gives the reason and docs/CARRIED-RULES.md the full text. Where it is new, it is marked **New**.

Child, not entity. A child is the unit of packaging and design: a whole entity, or a bounded part of one, decided in Structuring (§5.3). Every step after Structuring speaks of children. Where a request covers one entity split into several children, their crossings are paired in Hand-offs like those between entities.

5.0 The stages: input, output, done-test

Each stage is one run state and closes on its gate. The last stage closes on T8, then T9. "Settled" and "clear" are defined in §5.8 (Done when). "Unresolved" has the meaning of Annex D item 12.

ID	Plain name	Run state	Gate(s)	Exact input	Exact output (contract or file)	Done-test	Human decision
S0	**Admit**	admitting	T0	The run Work Item from the Requesting Authority's Cell, whose sources holds one approved Design Request reference. The connection grant of each adapter. Each adapter at its pinned snapshot	**Input Pack P0** (contracts/input-pack.schema.json), with its pack report. The run record, opened. The **run preview** (Annex E.13), with the run depth, the authoring of S1 and S5, and the catalogue version or	**T0** (Annex C.2): conditions (a) to (j).	Requesting Authority: approves the Design Request in Source Lifecycle and opens the run. Seat over [ORG]AIOC-SL: signs the connection. TOM Design Lead: acknowledges the

ID	Plain name	Run state	Gate(s)	Exact input	Exact output (contract or file)	Done-test	Human decision
				(v0.4.0: ADP-SL only). The engine register, read at the tool gateway. The pins of contracts/PINS.json, among them the reference-model catalogue's (Annex E.19).	unavailable. The RP0 review record (run record).		run preview and, where only a reduced level is possible, accepts it within the TM Cell's permission. Requesting Authority: states the run depth in the Design Request (DRQ-01 field 9). TOM Design Lead: where two authors at S5 would lower its review, accepts single authoring there within the TM Cell's permission (Annex G.7).

ID	Plain name	Run state	Gate(s)	Exact input	Exact output (contract or file)	Done-test	Human decision
S1	**Frame**	framing	T1	The Input Pack version named by each step (P0 or later). The Design Request's atoms. The request kind profiles (docs/ASKING-CATALOGUE.md §3b).	The **Design Brief frame**, reconciled from two drafts written blind by two authors of two families (Annex G), with its Gate A record (field list: docs/CARRIED-RULES.md §4.1 items 1–12 and 14–20; JSON Schema design-brief: build item). The **party register** (Annex E.3). The **strategy frame** and the **current-state activity profile** (release parts; Annex H.1, H.8). The do-not-ask and do-not-use-as-fact lists. The lens record (run record, Annex E.14). The two locked drafts and the **difference register** (run records; Annex E.18). The **frame decision pass**:	**T1**: both drafts locked before the compare step, and every material difference disposed under Annex G.5; RP1b and RP1 settled clear before any row was sent; Gate A recorded and re-assessed at RP1; party register reviewed at RP1; the non-delegable rows decided (none "don't know"; weak-approval signal cleared); the strategy frame row decided, with purpose and values cited from a source	Requesting Authority: scope, exclusions, roster (committed or context), contradictions touching them; the reading meant, where two readings differ (reading A, reading B, both readings); the strategy frame (Annex H.1). Cell carrying [ORG]HOAC-GOV: purpose and values, where no source in force states them. Delegable rows may be answered by delegation (recorded delegated, not decided).

ID	Plain name	Run state	Gate(s)	Exact input	Exact output (contract or file)	Done-test	Human decision
					needs requests carrying non-delegable decision rows (scope, exclusions, the entity roster, contradictions touching them, and each reading of the request that the two drafts differ on and the sources do not settle) and delegable confirmation rows (understanding check, "found, not said", load-bearing assumptions, obligations). Or a **Readiness Report** for the whole request.	in force or signed (Annex H.1); the activity profile compiled, or no observation (Annex H.8).	
S2	**Structure**	structuring	T2	The decided frame. The current pack version. The party register.	The **Structure Proposal**, then the decided structure and first slice (field list: docs/OUTPUT.md §5 and CARRIED-RULES.md §6; JSON Schema structure: build item), child IDs CHILD-.... The party register extended with the sibling children.	**T2**: orphans 0; exclusion violations 0; every child passes the four legs; RP2 settled clear before any row was sent; each child row and the first-slice row decided; the party register extended with the sibling children.	Requesting Authority: each child; the first slice.

ID	Plain name	Run state	Gate(s)	Exact input	Exact output (contract or file)	Done-test	Human decision
S3	**Child briefs**	child briefing	T3	Per child of the slice: its decided structure row; the decided frame; the current pack version.	Per child: a **child brief** (its Gate A record and floor; its criteria; the contradictions decided for it; JSON Schema design-brief, child part: build item), or a	**T3**, per child: Gate A recorded and re-assessed at RP3; floor holds; RP3 settled clear before any row was sent; every non-del	Requesting Authority: the child's criteria; contradictions touching the child.

ID	Plain name	Run state	Gate(s)	Exact input	Exact output (contract or file)	Done-test	Human decision
					child **Readiness Report** section.	egable row decided, none "don't know"; or a Readiness Report section.	
S4	**Package**	packaging	T4	Per child: its child brief; the pack atoms of the current version (Packaging selects atoms; it creates none); the party register (quarantined, read only by code steps); the decided constraints.	Per child: the **child package** (six sections and the integrity line; field list CARRIED-RULES.md §7.1; JSON Schema child-package: build item); the **packaging ledger** (packaging-ledger: build item); the marker-to-identity map, the withheld-context record and the packaging record (run records).	**T4**: G1–G5 pass for every child package; RP4 settled with no unresolved MATERIAL finding.	TOM Design Lead: rules on contested MATERIAL findings. Requesting Authority: only the choice on a material finding notice.
S5	**Design**	designing	T5	Per child: the child package; and the reference-model catalogue at the run's pinned version, a method and never evidence (Annex B.5). Nothing else.	Per child: two **design ledger** drafts, written blind by the Designer and the Second Designer and locked (one, under Annex G.7); the **difference register** (run record; Annex E.18); the reconciled and revised **design ledger** (field list CARRIED-RULES.md §8.2; JSON Schema design-ledger: build item), with each capability's delivery model and maturity and each value stream's criticality (Annex H.2–H.4); Output A (TOM Blueprint) and Output B (Rationale and Evidence Brief), rendered by code; the revision record.	**T5**: schema and code checks (CARRIED-RULES.md §8.3) pass; the difference register complete, every material difference disposed under Annex G.5; RP5 settled with no unresolved MATERIAL finding; the step is complete, or incomplete-design routed as a needs request or a WITHDRAW choice.	As S4.

ID	Plain name	Run state	Gate(s)	Exact input	Exact output (contract or file)	Done-test	Human decision
S6	**Hand-offs**	hand-offs	T6	The decided design ledgers of the slice, and those of any carried children; their declared boundary registers; the marker-to-identity maps; the withheld-context records; the party register; the deferred-boundary register; the decided Design Brief and structure.	The pairing table (run record; Annex E.6a). The **seam findings**, the **hand-offs register**, the **counterparty register**, the **legal and regulatory perimeter** and the **deferred-boundary register** (release parts; Annex E.4–E.7a). The **role register**, with its role findings (release part; Annex E.17), compiled by code. The **outside-party journeys** (release part; Annex H.5).	**T6**: every declared crossing is paired or is a seam finding; every journey moment is a crossing of its entry, and every crossing that delivers a capability is paired, registered or deferred (Annex H.11 checks 2 and 4); every seam finding is resolved upstream, decided or registered; the role register compiled by code, and no role finding open; RP6 settled with no unresolved MATERIAL finding.	Requesting Authority: seam decision rows; role rows (two names, one role?); notice choices. TOM Design Lead: rulings.
S7	**Walk**	walking	T7	The **walk input** (Annex E.8) and the **blind brief** (Annex E.9), both compiled by code; the version of the standard situation library and the request kind's situation pack. The run depth (Annex F), which sets the situation budget above its floors.	The sealed situation set; the trace; the **finding register** (field list CARRIED-RULES.md §10.3 and §10.6; JSON Schema finding-register: build item); the **walkthrough report**.	**T7**: situations sealed before the walk; the mix met or its shortfall declared; RP7 settled with no unresolved MATERIAL finding.	Requesting Authority: notice choices. TOM Design Lead: rulings.

ID	Plain name	Run state	Gate(s)	Exact input	Exact output (contract or file)	Done-test	Human decision
S8	**Decide and release**	deciding; end states released, then filed	T8, then T9	Every reviewed part of S1–S7, by hash; every pack version; the settled review records and rulings of S0–S7.	The **quality checks** (Annex E.11), generated by code from the settled review records, and extended after RP8. The TOM Design Lead's attestation of T0–T7. The **Decision Package** (field list CARRIED-RULES.md §12.1; JSON Schema decision-package: build item), which states in plain words the run depth, the general patterns relied on, and the role register; the **value case** (release part; Annex H.6), compiled by code; its decision rows, each element row with its proposed priority (Annex H.7); the decision references (admitted as pack versions). Then the **TOM Release** manifest (contracts/tom-release.schema.json) and its parts; the release lint report (run record, Annex E.15); the **filing receipt** (contracts/tom-filing-receipt.schema.json); the **release notice** to the next stage (contracts/release-notice.schema.json; §5.9 step 8a).	**T8**: RP8 settled clear before any row was sent; every element row (each entry of the role register included) and the final status decided; each element row decided ACCEPT or CORRECT carries its priority, and the value case row is decided (Annex H.6, H.7); purpose-and-values rows carry the GOV Cell's gateway signature. **T9**: the manifest is content-addressed; every element traces to an atom of a pack version of this run or of a carried release; every pattern a ledger cites resolves in the catalogue pinned for the run, by version and hash; the role register's entries re-compile from the frozen parts and the decision references; the value case and the	Requesting Authority: each element (ACCEPT, CORRECT, REJECT) and the final status. Cell carrying [ORG]HOAC-GOV: purpose-and-values rows, signed at the gateway. TOM Design Lead: the attestation of T0–T7.

ID	Plain name	Run state	Gate(s)	Exact input	Exact output (contract or file)	Done-test	Human decision
						activity profile re-compile from the frozen parts and the decision references (Annex H.11 check 5); the release lint passes; RP9 settled with no MATERIAL finding; the release is filed and reads back by hash.	

Not stages: services and rules used by every stage.

Name	What it is	Input	Output	Done-test
Asking (capability Needs Requests, §5.2)	A service of the Chief of Staff. Every stage raises needs candidates; the Chief of Staff asks the Chief of Staff of [ORG]AIOC-SL. Answers come back through the Input socket as new pack versions.	Needs candidates, each with its leverage proof	Needs requests (contracts/needs-request.schema.json), one Missing Report per pass (contracts/missing-report.schema.json), status messages (contracts/needs-status.schema.json)	Each pass passes review point RPa before it is sent. Each request ends answered by reference, or with another outcome of the needs-answer contract, and its sub-item is done only when the acceptance test of §4.3 passes.
Review Panel rule (capability Independent Review, §5.8)	The rule that closes every stage: several engines review each locked output and debate.	A review packet per review point	A review record (run record) and a verdict	§5.8, Done when.

Name	What it is	Input	Output	Done-test
Two authors (Annex G)	At S1 and S5, two authors of two families write the output blind; code compares the two drafts; a reconciler makes one version	the stage's inputs, given to each author	two locked drafts; the difference register (run record); one reconciled output	Annex G, Done when
Run depth (Annex F)	How finely the run designs, and how much it checks above the floors	DRQ-01 field 9, read at P0	the depth, fixed for the run	T0 condition (i)
Labels (Annex B)	Every atom and every material statement carries one label.	—	—	Annex B.

Run states (Annex C.1).

- States, in order: admitting (S0) → framing (S1) → structuring (S2) → child briefing (S3) → packaging (S4) → designing (S5) → hand-offs (S6) → walking (S7) → deciding (S8) → end states released → filed.
- Endings other than a release: closed: not admitted (T0 failed; the run Work Item is returned with each failed condition by name); closed: readiness report; closed: held report; closed: non-response; closed: rejected; closed: withdrawn by the Requesting Authority.
- Flags on any state, each shown with its age: waiting; waiting: ruling; held: material finding; held: connection revoked; held: no reviewers; held: review budget; held: engine ceiling.
- No flag releases the run slot. Every ending releases it.
- Per-child stages (S3, S4, S5) run per child. The run state is the earliest stage any child of the slice is still in.

One word per meaning.

- **Review Panel**: the rule of §5.8, and the set of engines drawn for one review. It is never the Panel.
- **Panel**: the AIOC's screen (§4.4).

- **Engine**: an AI model as served through the tool gateway. **Family**: its lineage.
- **Operating model**: only what TOM designs. In engine text "the model" means the operating model; "model" never names an engine.
- **Crossing**: one child's declared boundary entry (seven fields, docs/CARRIED-RULES.md §7.1). **Hand-off**: a clean pair of crossings between two children, recorded in the hand-offs register. A child's crossing with an outside party is recorded in the counterparty register.
- **Job**: the per-child run of a per-child stage ("one child per job"). **Run**: the whole design run (one Work Item).
- **Release file** in reader text; **part** in the manifest and in engine text.
- **Author**: an AI Cell whose engine produced an output (§5.8 rule 3). **Two authors**: two such Cells, bound to engines of two families, writing one output blind to each other (Annex G).
- **Draft**: one author's locked version of the frame (S1) or of a child's design ledger (S5). **Difference**: one item on which the two drafts differ (Annex E.18). **Reconciler**: the Cell that makes one version from two drafts (Annex G.1).
- **Run depth**: how fine the model is, light, standard or deep (Annex F).
- **Pattern**: an entry of the reference-model catalogue of the 49GAION Library; a method, never evidence (Annex B.5). In reader text: "general pattern".
- **Role register**: the release part that lists every role-type once (Annex E.17). It lists roles, never positions: the Position Register belongs to the next stage (Naming Standard §3).
- **Strategy frame**: the organisation's strategy on one page, as its sources state it or as it decides it (Annex H.1). **Value case**: per outcome, its measures, baselines and targets, compiled by code (Annex H.6). **Activity profile**: how the work runs today, by role-type and process, from approved exports only (Annex H.8). **Journey**: the moments where one outside party meets the organisation, in order (Annex H.5).

5.1 Request Reconstruction Definition

Field	Content
Name	Request Reconstruction
What it is	The reconstruction of what the organisation is asking for, **from its sources alone**: who asks and with what authority to commit, what is in and out, what exists today, what should be different, how anyone would know, what limits apply, and what is unknown or contested. It ends in a **Design Brief**. The brief has a frame for the whole request (stage S1), decided first, and then one child brief per decided child (stage S3). A child whose floor holds gets its child brief; a child whose floor fails gets a **Readiness Report** section instead. The frame is written by two authors, blind, and reconciled (Annex G).
Purpose	Understand the request as completely as the sources support, and no further, before any burden is put on a person.
Scope	Does: read, cite, read through lenses, reconstruct, label, find contradictions and gaps, state the request kind, write the party register, record Gate A, raise needs. Does not: design the solution; choose tools; write a success criterion; settle a contradiction; treat an inference as confirmed; narrow the request alone; answer its own Gate A independently (the review at RP1 does that, §5.8).
Inputs	The current Input Pack version (§4.3), named by each step; answers to earlier needs requests, as later pack versions; the request kind profiles (docs/ASKING-CATALOGUE.md §3b); for the child briefs, the decided structure (§5.3).

Field	Content
Outputs	The **Design Brief** (docs/OUTPUT.md §4), versioned. The **strategy frame** (Annex H.1) and the **current-state activity profile** (Annex H.8), release parts. The **party register** (Annex E.3). The **lens record** (Annex E.14). The two locked frame drafts and the **difference register** (Annex E.18). The **do-not-ask list**, bounded to: (a) needs candidates raised by a Pod in this run; (b) the fields of the published contracts this run must fill (the Design Brief sections, the explainer templates in play); (c) needs answers already received. Each entry has the answering citation; the list is never an unbounded enumeration. The **do-not-use-as-fact list** (weak or inferred items). The **frame decision pass** (docs/CARRIED-RULES.md §4.5) and needs requests (§5.2). Where no frame can be decided at all, a **Readiness Report** for the whole request (docs/OUTPUT.md §4, last paragraph).
Done when	S1: gate T1. S3: gate T3, per child (§5.0).
Limits	Citation or refusal: every answer carries a source reference, or it is refused and recorded as unknown. No new world facts and no outside research. No claim may cite a source the read plan marks as not opened. An unknown or contested item closes only on a cited source; a silent closure fails the run. An inferred item stays inferred, and keeps raising the question, until the organisation confirms it through Source Lifecycle (sticky inferred, Annex B.1). "The organisation wants..." must trace to a source, never to inference; so must every element of the strategy frame (Annex H.1). Which of two disagreeing sources governs is settled only by a precedence rule a held source states, cited, or by a non-delegable decision of the Requesting Authority for this brief (Annex B.2). This AIOC never writes a success criterion; it checks a criterion's admissibility (Annex C.3). Frame and Child briefs read the Input Pack only. The Structure Proposal, the child package and the design ledger are consumer contracts and never enter reconstruction (docs/CARRIED-RULES.md §2.6). The two frame drafts are reconciled only under Annex G.5.
Sources	49GAION Operating Protocol v1.5.0 §8 (provenance), §13 items 4 and 6 (unreleased v1.6.0 row); Standard v1.5.0 §3.2 items 1 and 8. Design basis: docs/DESIGN-BASIS.md §1, §2; rule texts: docs/CARRIED-RULES.md §1-§4, §2.7, §2.8, §4.5

Target Operating Model

Field	Content
Status	draft

Process (SOP)

Field	Content
Trigger	Gate T0 has passed (Annex C): an admitted run request carries an approved Design Request, and Input Pack P0 is settled at RP0.
Steps	see the numbered steps below the table
States	Frame: forming → waiting on sources → audited → awaiting decision → frame decided. Child brief: forming → audited → awaiting decision → decided · or readiness report. Either: stopped: non-response. The Chief of Staff moves the states; only a recorded decision moves a brief to decided.
Failure and recovery	Two consecutive rounds with no material improvement (docs/CARRIED-RULES.md §2.4) stop the rounds. At the cap of 12 rounds the step stops with stop_reason = round-cap; more rounds need new sources. After the configured number of passes past their waiting deadline (§15) with non-delegable rows unanswered, the step stops with stop_reason = non-response. It then emits a Readiness Report, or the narrowest honest brief built only from non-delegable rows already decided. At any stop, each residue item is marked safe-to-carry or blocking. Blocking residue, or residue holding a standing, outcome or startable-criterion item, forces a Readiness Report. stop_reason is one of sufficiency, diminishing-returns, round-cap, floor, non-response. A frame that cannot be decided at all ends the run in a Readiness Report for the whole request. A frame whose parent floor fails does not: Structuring runs, and the floor applies per child. Two failures on the same step stop the step and go to the TOM Design Lead.
Records	Run record: read plan, round log, Working Context, lens record, party register, Gate A records, brief versions, review records, decision references.
Cadence	per run

Field	Content
Measures	Burden spared: the size of the bounded do-not-ask list. Needs sent, with leverage proofs. Unknowns in and out: closed, transformed, survived. Label distribution. Lenses retained, guard and dropped. No target is set.

The frame (S1).

Steps 1–9 are run twice, blind (Annex G.2): by the Request Analyst, and by the Second Request Analyst on an engine of the other author family, each with its own Source Proxy rounds. Each output of steps 1–9 is that author's own.

0. **Chief of Staff (runs the activity-profile code step)**: before the authors start, and again on each pack version that admits an export of records of the work, compiles the current-state activity profile, or no observation (Annex H.8) → activity profile (evidence: hash; mechanically-verified). Both authors read it as present state.
1. **Request Analyst**: declares the **read plan** before reading: what is read in full, sampled or not opened, and why the unopened material is unlikely to change a decision; the selection basis carries its own label; the organisation dictionary is opened wherever Source Lifecycle serves one (Annex H.9). The socket atomises each source the plan opens, as a new pack version, where the corpus is above lazy_atomising_threshold → read plan (evidence: run record entry and hash).
2. **Source Proxy**: screens each source's Source Lifecycle status. Recency is never authority; presence in the sources is never adoption; a precedence between two sources is admitted only when a held source states it, cited. The pack's live_version_conflicts enter as candidate contradictions → currency screen (evidence: run record).
3. **Request Analyst ↔ Source Proxy**: internal rounds. The Proxy answers only from the sources, with a citation, or refuses. The Analyst may ask about situation, history, constraints and intent found in the sources. It may never ask about solutions, design preferences, tools or success criteria → round log (evidence: run record).

4. **Request Analyst**: the lens pass (docs/CARRIED-RULES.md §2.7), with at most the lens cap of the run depth (Annex F) → lens record (evidence: run record).
5. **Request Analyst**: builds the Working Context: what exists today; who does what; the contradiction register; what was expected and not found; unclear authority; the currency screen; every row labelled (Annex B). Retained lens findings enter only with their citations → Working Context (evidence: run record).
6. **Request Analyst**: states the request kind with its reason (CARRIED-RULES.md §2.8), and writes the party register (Annex E.3) → party register (evidence: run record).
7. **Request Analyst**: reconstructs the frame. It covers: the requesting authority per entity, each entity committed or context, with its veto-holders; the request as interpreted; scope in; named exclusions; present state; outcomes, each stated as a difference from today that two independent readers would judge the same; criteria only as the sources state them, each checked for admissibility (Annex C.3); constraints; assumptions, each with what breaks if false; unknowns, carried or blocking; contradictions; obligations; open items; the request kind; the honesty ceiling; and the strategy frame, each element cited or not given (Annex H.1) → frame draft (evidence: run record and hash).
8. **Request Analyst**: records **Gate A** for the whole request: the eight questions of Annex C.3, each answered yes or no with a reason → Gate A record (evidence: run record).
9. **Request Analyst**: applies the **decision-leverage test** to each item the sources cannot answer: does any scope boundary, constraint, criterion or design decision change with its value? If not, the item is carried as an assumption and not asked. If so, it becomes a needs candidate (§5.2), ranked by breadth, irreversibility and cost of being wrong. An element of the strategy frame that no held source states is always a candidate, asked once through EX-10 (Annex H.1, item 3) → needs candidates with leverage proofs (evidence: needs log).

9a. **Chief of Staff (runs the compare code step)**: once both authors' drafts of steps 1–9 are locked by hash, aligns them and writes the difference register (Annex G.4; Annex E.18) → difference register (evidence: hashes;

mechanically-verified). 9b. **Frame Reconciler**: in a fresh context, with only the two locked drafts, the difference register, the atoms they cite and the rules, reconciles them into one frame, party register, do-not-ask and do-not-use-as-fact lists and set of needs candidates, under Annex G.5, recording each disposition; a reading of the request that only the organisation can settle becomes a row of the frame decision pass (step 12) → reconciled frame (evidence: run record).

10. **Review Pod**: the process audit, review point RP1b (§5.8), over both authors' round logs and lens records. A MATERIAL verdict stops the step, and the author whose rounds hold the defect re-forms its draft from its last clean round; steps 9a and 9b then run again → RP1b record (evidence: the review record).
11. **Review Pod**: the frame review, review point RP1 (§5.8), with the question set of CARRIED-RULES.md §11.1. It sees only the pack atoms the frame cites, the reconciled frame with its Gate A record, the party register, the two locked drafts and the difference register, never the rounds. It re-assesses every Gate A answer and the floor on its own → RP1 record (evidence: the review record).
12. **Request Analyst**: runs only after RP1b and RP1 have settled clear. It sends the frame decision pass (CARRIED-RULES.md §4.5), through the Chief of Staff (§5.2): non-delegable rows for scope, exclusions, the roster and contradictions; delegable confirmation rows for the understanding check, findings not said, load-bearing assumptions and obligations; a non-delegable row for each reading of the request that step 9b could not settle (reading A, reading B, both readings); one non-delegable row for the strategy frame, and the purpose-and-values rows where no source in force states them (Annex H.1) → frame decision pass (evidence: needs log).

The child briefs (S3), after the structure is decided (§5.3).

13. **Request Analyst**: for each decided child, records Gate A and the floor on that child (Annex C.3), and checks each criterion the sources state for admissibility (Annex C.3) → child brief draft and Gate A record (evidence: run record).
14. **Review Pod**: the child brief review, review point RP3 (§5.8) → RP3 record (evidence: the review record).

15. **Request Analyst**: after RP3 has settled clear, sends the child's non-delegable rows, through the Chief of Staff (§5.2): its criteria (at least one startable, admissible criterion), and each contradiction touching the child that no held source settles. A child whose floor cannot hold gets a Readiness Report section; the request resolves as a paired set of child briefs and child Readiness Report sections → child decision rows, or Readiness Report section (evidence: needs log; run record).

Role card

Field	Source Proxy	Request Analyst	Second Request Analyst	Frame Reconciler
Name	Source Proxy	Request Analyst	Second Request Analyst	Frame Reconciler
Position and seat	Source Proxy; accountable seat the TOM Design Lead ([ORG]HOAC-ACS)	Request Analyst; same	Second Request Analyst; accountable seat the TOM Design Lead ([ORG]HOAC-ACS)	Frame Reconciler; same
Performed by	an agent ([ORG]CELL-TSP)	an agent ([ORG]CELL-TRA)	an agent ([ORG]CELL-TSRA), on an engine of the other author family	an agent ([ORG]CELL-TFR), in a fresh context, on an engine of one of the two author families
Purpose	answer only from the sources, with a citation, or refuse	reconstruct the request into the Design Brief; derive and read the lenses; write the party register; raise only the needs that pass the leverage test	write the frame a second time, blind to the Request Analyst's draft	reconcile the two locked frame drafts into one, under Annex G.5

Field	Source Proxy	Request Analyst	Second Request Analyst	Frame Reconciler
Authority limits	alone: answer with a citation, refuse; never: answer without a citation; add a fact	alone: draft, label, derive and read lenses, state the request kind, write the party register, raise candidates; never: ask about solutions, preferences, tools or criteria; close an unknown without a source; let a lens finding enter the Working Context without a citation; send a decision row before RP1b and RP1 have settled clear	alone: as the Request Analyst, for its own draft; never: see the other author's draft; anything the Request Analyst may never do	alone: dispose each difference under Annex G.5, with reasons and quotes; correct the code's alignment with a quoted reason; never: add what neither draft holds; choose by preference; raise a label, standing or Gate A answer; settle what only the organisation can settle; write a verdict or a verification label
Write location	the round log	the Working Context, the brief drafts, the lens record, the party register, needs candidates	its own draft, round log, lens record, party register and needs candidates	the reconciled frame and the dispositions of the difference register
Reports to	the Chief of Staff	same	the Chief of Staff	same
Escalation	to the Chief of Staff in one line; a stop goes to the stop door	same	to the Chief of Staff in one line; a stop goes to the stop door	same
Tools and access	none beyond its inputs; no file, search or web tool; access only through the Panel; no credential in any instruction	same	none beyond its inputs; no file, search or web tool; access only through the Panel; no credential in any instruction	none beyond its inputs; no file, search or web tool; no credential in any instruction

Field	Source Proxy	Request Analyst	Second Request Analyst	Frame Reconciler
Deputy and cover	none; on failure the step halts and the Chief of Staff raises it	same	none; on failure the step halts and the Chief of Staff raises it	same
Fixed test and score	TEST-TOM-TSP v5 (cases 4, 5, 14, 92)	TEST-TOM-TRA v5 (cases 8, 14, 15, 28, 73-75, 83, 91, 92, 109-111, 123)	TEST-TOM-TSRA v5 (cases 8, 14, 15, 73-75, 83, 91, 92, 109-111, 123)	TEST-TOM-TFR v5 (cases 93, 94)
Card version and status	v0.4.0 · draft	v0.4.0 · draft	v0.4.0 · draft	v0.4.0 · draft

Fixed test

TEST-TOM-TRR v5: TEST-TOM v5, cases 4, 5, 8, 14, 15, 28, 73-76, 83, 91-93, 109-111, 121 and 123 · result NOT_RUN

5.2 Needs Requests Definition

Field	Content
Name	Needs Requests
What it is	The only way this AIOC learns anything a source does not already say. A needs request asks the **Chief of Staff of [ORG]AIOC-SL** for one thing: a document, an explainer, a confirmation, a status, an export or a decision. That Chief of Staff gets it from the position that can give it, has it approved and recorded as a source, and answers by reference. With every pass goes a **Missing Report**: everything still missing for the operating model, by kind, in plain words. The full contract is in docs/INTAKE-WITH-SOURCE-LIFECYCLE.md; the forms and explainer templates are in docs/ASKING-CATALOGUE.md. Answers come back through the Input socket as new pack versions (§4.3).
Purpose	Get what the design needs with the least burden on people, and without ever treating spoken words as a source.
Scope	Does: write, batch, send, track and withdraw needs requests; send the Missing Report; receive answers and status messages by reference. Does not: contact a person; answer a request itself; draft the organisation's content; choose the person who answers; decide.
Inputs	Needs candidates from any Pod, each with its leverage proof.
Outputs	Needs requests, each a sub-item of the run Work Item to [ORG]AIOC-SL (contracts/needs-request.schema.json); confirmation rows carry their row_purpose; one Missing Report per pass (contracts/missing-report.schema.json); withdrawn and superseded status messages (contracts/needs-status.schema.json); the needs log.
Done when	Each pass: RPa settled clear before it is sent. Each request: answered by reference, or ended with another outcome of the contract; its sub-item is done only when the acceptance test of §4.3 passes.

Field	Content
Limits	A need is sent only when no held source answers it; its minimality proof lists the references searched, at least one. One decision per row. At most the configured rows per pass (§15, default 12). Each row declares the rows it depends on. The request carries the raising Pod's own wording verbatim beside the plain-language question. Silence is never an answer. A general approval, or approval of a whole pass, decides no row. Every pass is reviewed at RPa (§5.8) before it is sent. **New.** A request is never sent to a person, and never names a person. Its routing is bound only for a non-delegable row: the code of the Requesting Authority's Cell, or, for a listed type, of the Cell that signs it. For any other row it may suggest the role the held sources name, with the citation, and the Chief of Staff of [ORG]AIOC-SL chooses who answers.
Sources	49GAION Operating Protocol v1.5.0 §4.1 (Work Item), §4.2 (statuses), §4.4 (bounce limit), §4.5 (done), §4.6 (waiting time and sub-items), §13 items 2 and 6. Design basis: docs/DESIGN-BASIS.md §1; rule texts: docs/CARRIED-RULES.md §5
Status	draft

Process (SOP)

Field	Content
Trigger	A needs candidate passes the leverage test in any Pod.
Steps	see the numbered steps below the table
States	A view over the one status list of Protocol §4.2. received (sent); in-progress (accepted by Source Lifecycle, or asked); rejected (returned by Source Lifecycle, with its reason; withdrawn by this AIOC, with the reason "withdrawn by requester"; or superseded by a new version); done (answered, declared absent, not applicable, declined or "don't know", set only when this AIOC's acceptance test passes). The run item shows needs-input while any needs request is open. The needs log also keeps the finer states accepted, asked, returned, withdrawn and superseded from the status messages.
Failure and recovery	A request returned as malformed is fixed and re-sent up to the configured bounce_limit (Protocol §4.4); past it, it goes to the TOM Design Lead. An answer that does not resolve the question becomes a new request, never a guess. An overdue request follows the waiting deadlines and escalation (Protocol §4.6). The item it blocks is carried as unknown, with its cost. A pass that RPa holds is not sent: the offending rows are withdrawn or rewritten by their raising Cell, and RPa runs again.
Records	The needs log: every request, its pass, its versions, its state history, its answer reference; every Missing Report; the RPa records.
Cadence	per pass; passes are sequenced so that the organisation never receives more than the configured rows at once
Measures	Needs per run; share answered; median age by type; answers that changed a design element. No target is set.

1. **Raising Cell**: drafts the candidate: type; the question in its own wording; why it matters; which element, decision or gate it blocks; what changes depending on the answer; what will be carried if no answer comes, and what that costs; the rows it depends on; the references searched → candidate (evidence: needs log).

2. **Chief of Staff**: checks the candidate against the do-not-ask list and the needs log, and drops a duplicate or an answered question → checked candidate (evidence: needs log).
3. **Chief of Staff**: writes the plain-language question for the organisation, keeping the original wording verbatim beside it, and picks the asking form (docs/ASKING-CATALOGUE.md); a confirmation row states its row_purpose → needs request (evidence: the request validates against its contract).
4. **Chief of Staff**: batches the pass: at most the configured rows, one decision per row, ranked by leverage. Non-delegable rows name their answering Cell (the Requesting Authority's, or for a listed type its signer's) and come first. An explainer counts as three rows toward the pass limit → pass (evidence: needs log).
5. **Chief of Staff**: writes the pass's **Missing Report** (docs/INTAKE-WITH-SOURCE-LIFECYCLE.md §8.1) → Missing Report (evidence: it validates, and its open needs equal the open entries of the needs log).
6. **Review Pod**: review point RPa on the pass (§5.8) → RPa record (evidence: the review record).
7. **Chief of Staff**: sends each request as a sub-item of the run Work Item, and the Missing Report with the pass, through the output port to [ORG]AIOC-SL → Work Item IDs (evidence: needs log).
8. **Chief of Staff of [ORG]AIOC-SL** (outside this AIOC): accepts or returns each request with a reason, and reports asked; obtains the answer; has it approved; records it as a source; answers by reference, or with declared absent, not applicable, declined, dont know or returned → needs status messages and needs answer (evidence: Source Lifecycle's serve log).
9. **Input socket**: admits the answer only by reference, through the acceptance test of §4.3, into a new pack version; a typed or pasted answer is refused → pack version (evidence: the pack's increment).
10. **Chief of Staff**: sets the sub-item done only when this AIOC's acceptance test passes on the answer (Protocol §4.5); updates the needs log; re-opens only the work that cites an affected atom; returns to undecided every row

declared as depending on a row answered correct or dont know → dependent work re-opened (evidence: run record).

11. **Chief of Staff**: withdraws a request the run no longer needs, or supersedes it with a new version, and tells Source Lifecycle, which stops asking → needs status (evidence: needs log).

Role card

Field	Content
Name	Chief of Staff (needs requests)
Position and seat	Chief of Staff; accountable seat the TOM Design Lead
Performed by	an agent ([ORG]CELL-COSTOM); the raising Cell drafts each candidate
Purpose	ask the organisation, through Source Lifecycle, only what the design needs and the sources cannot answer
Authority limits	as in §4.2; never: name a person; choose who answers a delegable row; send a pass before RPa has settled clear, or a decision row before the review of its stage has settled clear (§5.8)
Write location	the needs log
Reports to	the TOM Design Lead
Escalation	as in §4.2
Tools and access	as in §4.2
Deputy and cover	none; the run halts
Fixed test and score	TEST-TOM-COSTOM v5 (cases 7-13, 28, 30, 75, 76, 109)
Card version and status	v0.4.0 · draft

Fixed test

TEST-TOM-COSTOM v5 (cases 7-13, 28, 30, 75, 76 and 109 for this capability)
· result NOT_RUN

5.3 Structuring

Definition

Field	Content
Name	Structuring
What it is	The proposal of how the decided frame breaks into **children**: each separately ownable, boundable and startable, **without designing what any child contains**. A child is a whole entity or a bounded part of one. Structuring also proposes the **first slice**, the children to design first; the proposal binds only as a decision of the Requesting Authority.
Purpose	Give each later step one clear, bounded piece of work, with nothing in scope dropped and nothing excluded smuggled in, so that even a request too large for one brief can be designed.
Scope	Does: propose children and a dependency order; check orphans and exclusions; record rejected alternatives; propose the first slice; extend the party register with the sibling children. Does not: design roles, processes, tools or criteria; dispatch work; approve its own proposal.
Inputs	The decided frame of the Design Brief: the entity roster with committed and context, scope, exclusions, unknowns and contradictions with their positions, and the honesty ceilings; the current pack version; the party register.
Outputs	The **Structure Proposal** (docs/OUTPUT.md §5) and its decision rows; after the decision, the **decided structure**; the party register, extended with the sibling children (Annex E.3).
Done when	Gate T2 (§5.0).

Field	Content
Limits	A child description holds subject, boundary in and out, and intended outcome, and nothing else. Every boundary cites the brief or a source. A child covering coordination between independent entities is named as relationships **between** them, never as a body above them. A contradiction touching a child's boundary, ownership or startability makes that child not startable pending decision. The dependency order is an ordering, never a dispatch. **New:** the first slice. It brings back a sequencing proposal that the earlier chain had removed, so it binds only as a decision row of the Requesting Authority. A release covers one slice; the next slice is a continuation run (§5.9, Slices and carry-forward).
Sources	49GAION Standard v1.5.0 §1.7 (composition); Operating Protocol v1.5.0 §4.6 item 2. Design basis: docs/DESIGN-BASIS.md §3; rule texts: docs/CARRIED-RULES.md §6
Status	draft

Process (SOP)

Field	Content
Trigger	The frame of the Design Brief is decided (gate T1). The parent floor need not hold: a request whose parent floor fails is structured, and the floor then applies per child (§5.1, steps 13-15).
Steps	see the numbered steps below the table
States	proposing → audited → awaiting decision → decided. A child the Requesting Authority rejects becomes a returned issue and blocks no sibling, except a sibling that declared a dependency on it, which becomes accepted, not startable. Either: stopped: non-response (§5.1, failure).
Failure and recovery	An orphan or an exclusion violation fails the run. A live dependency cycle fails the run: it is broken by cited re-scoping, or every child in it is marked not startable. The internal loop stops after 8 rounds. Two failures on the same step stop the step and go to the TOM Design Lead.
Records	Run record: the proposal versions, the checks, the RP2 record, the party register's extension, the decision references.
Cadence	per run
Measures	Children proposed; orphans caught; exclusion violations (must end at 0); alternatives considered; rounds used.

1. **Work Structurer**: derives the axes of division from the decided frame and the sources, each cited → axes (evidence: run record).
2. **Work Structurer**: proposes children and applies the **four-leg test** to each, answering yes or no with a reason. **Ownable:** one accountable owner role could hold it. **Boundable:** its in and out fit in three sentences without "everything" or "various". **Startable:** work could begin from the brief and this proposal alone. **Interfaces nameable:** what crosses to and from each sibling is listed. A child failing a leg merges upward. Of the sets where every child passes, the smallest wins → children with leg answers (evidence: run record).

3. **Work Structurer**: a single child equal to the whole request is admitted only with an attempted-split table: every axis tried, and why each fails the four legs. A vague "inseparable" note fails the run → split table where needed (evidence: run record).
4. **Work Structurer**: runs the orphan check: every in-scope item maps to at least one child, and the mapping table is part of the output. It also runs the exclusion check: no child crosses a named exclusion, and the matrix is shown → checks (evidence: run record).
5. **Work Structurer**: records at least one rejected alternative, with why → alternatives (evidence: run record).
6. **Work Structurer**: writes the dependency order (an ordering note, never a dispatch) and the proposed first slice, and extends the party register with the sibling children → ordering note; party register extension (evidence: run record).
7. **Review Pod**: review point RP2 (§5.8) → RP2 record (evidence: the review record).
8. **Chief of Staff**: after RP2 has settled clear, sends one non-delegable decision row per child, and one for the first slice (§5.2) → decision rows (evidence: needs log).

Role card

Field	Content
Name	Work Structurer
Position and seat	Work Structurer; accountable seat the TOM Design Lead
Performed by	an agent ([ORG]CELL-TWS)
Purpose	propose the smallest set of children that each pass the four legs, with nothing orphaned and nothing excluded
Authority limits	alone: propose, check, record alternatives, extend the party register with sibling children; never: design a child's content; approve its own proposal; drop a scope item; dispatch work
Write location	the Structure Proposal; the party register's extension
Reports to	the Chief of Staff
Escalation	to the Chief of Staff in one line; a stop goes to the stop door
Tools and access	none beyond its inputs; no file, search or web tool; no credential in any instruction
Deputy and cover	none; on failure the step halts
Fixed test and score	TEST-TOM-TWS v5 (TEST-TOM v5, cases 16, 42, 83)
Card version and status	v0.4.0 · draft

Fixed test

TEST-TOM-TS v5: TEST-TOM v5, cases 16, 42 and 83 · result NOT_RUN

5.4 Entity Packaging Definition

Field	Content
Name	Entity Packaging
What it is	For one decided child, the preparation of the **child package**: the only thing the model designer may see. It selects the child's own evidence from the atoms of the Input Pack, declares every boundary crossing as structured data, and replaces every counterparty's identity with a stable opaque marker.
Purpose	Let one child be designed on its own evidence, with every crossing declared and nothing of the far side leaking in.
Scope	Does: select the child's atoms; filter; declare crossings; anonymise; preserve contradictions. Does not: create an atom; design the child; resolve a contradiction; rank, rate, summarise or paraphrase evidence.
Inputs	One decided, startable child row and its child brief; the pack atoms of the current version; the party register (quarantined: read only by the code steps); the decided constraints.
Outputs	The **child package**, in exactly six sections: the child row; the child's own evidence, verbatim; the applicable constraints, verbatim; the **declared boundary register**; the preserved unknowns, contradictions and evidence ceiling, with a completeness flag (complete, partial or thin); the output-level statement. Three records are kept away from the designer: the marker-to-identity map; the withheld-context record (open only to review and Hand-offs); and the packaging record. The package is a run-workspace artefact, not a release part (docs/OUTPUT.md §2).
Done when	Gate T4, per child (§5.0).

Field	Content
Limits	No far-side content, ever. Identity removal is the default. Disclosure is allowed only in four declared cases: (a) the requesting authority; (b) a named regulatory regime whose concrete internal requirements for the child are quoted in the child's own evidence (the regime is disclosed, never the regulator or any other organisation); (c) the child's own legal identity; (d) a named platform or vendor imposing internal requirements through the child's own evidence. Each disclosure is declared in the packaging record as "identity disclosed because <case> + <atom reference>". Keeping one counterparty recognisable across crossings is never a ground for disclosure: the stable marker does that. Every judgement carries a reason code, its references, a confidence, and a "could be wrong because" note where it is inferred. One child per job. Rules 1-12 in full: docs/CARRIED-RULES.md §7.
Sources	49GAION Operating Protocol v1.5.0 §8; Standard v1.5.0 §3.2 item 8 (outside text is data). Design basis: docs/DESIGN-BASIS.md §4; rule texts: docs/CARRIED-RULES.md §7
Status	draft

The seven fields of each crossing in the declared boundary register:

- a stable marker (BX-n; provisional BXP-n for an evidenced party missing from the party register);
- direction (inbound, outbound, two-way);
- what crosses, in the child's own words;
- the child-side obligation or trigger;
- counterparty class (regulatory, commercial-counterparty, group-internal, financial-infrastructure, technology-platform, public, unresolved-class);
- provenance (stated, inferred, unresolved), with a confidence;
- optional criticality, only as citations to the child's own evidence.

Each field carries its own basis, or unknown.

Atoms come from the Input Pack. The Input socket cuts every source into atoms and labels each once (§4.3). Packaging selects atoms for a child and creates none. Atom text is resolved from Source Lifecycle into the run workspace only, and never edited. Records and releases keep atom IDs and hashes (§7).

Process (SOP)

Field	Content
Trigger	A child is decided and startable, and its child brief is decided (gate T3).
Steps	see the numbered steps below the table
States	packaging → gated → reviewed → released to model · or incomplete-package (with the blocking defect and its owner) · or atom-split-needed (raised as a need).
Failure and recovery	A malformed input goes back to its owner and is never repaired here. Two failed schema validations on the same step stop the job as incomplete-package and go to the TOM Design Lead. A mixed atom (far-side content remaining after substitution) is excluded and raised as atom-split-needed, never edited. If the exclusion removes evidence core to the child, the need is raised rather than shipping a thin package that pretends to be enough. A source that cannot be atomised is refused at the socket as not atomisable, and the need for a cleaner version is raised there (§4.3). A MATERIAL review finding goes to bounded repair (§5.8).
Records	The packaging ledger, the gate report, the RP4 record and the three records, in the run record (atom IDs and hashes only, §7).
Cadence	one job per child
Measures	Crossings declared; atoms withheld; candidates adjudicated; gate results; review findings.

1. **Chief of Staff (runs the prepackaging code step)**: over the pack atoms in the child's scope, builds one stable marker per party of the party register, with its aliases, abbreviations and role descriptors; replaces identities in

every atom, longest match first; then sweeps for crossing candidates: hand-off, receipt, notification, approval and reporting patterns, and every marker mention → marker store, class-only registry, candidate list, marker-to-identity map (evidence: code-step log and hashes).

2. **Packager**: sees only the markerised atoms, the class-only registry, the candidates, the child row and the constraints. It decides each atom's membership (in, out, withheld). It adjudicates **every** candidate as crossing, not-crossing or unresolved, with reasons and references. It selects and declares; it never retypes evidence → packaging ledger (evidence: run record).
3. **Chief of Staff (runs the package-assembly code step)**: assembles the package and the three records from the ledger and the store, pulling atom text by ID → package (evidence: hashes).
4. **Chief of Staff (runs the gate code step)**: runs gates **G1-G5** (docs/CARRIED-RULES.md §7.3): G1, no identity, alias or descriptor appears except under a declared disclosure; G2, markers and register match; G3, closed vocabularies and complete fields; G4, every atom is byte-identical to its source after marker substitution; G5, every candidate is adjudicated, with reason and references. Any finding blocks release → gate report (evidence: mechanically-verified).
5. **Review Pod**: review point RP4 (§5.8). The engines see the markerised atoms, the child row, the package, the packaging ledger, the gate report and the rules. They look for far-side content left after substitution, semantic identity leakage, evidenced crossings not declared, and atoms of the child's own withheld → RP4 record (evidence: externally-audited).

Role card

Field	Content
Name	Packager
Position and seat	Packager; accountable seat the TOM Design Lead
Performed by	an agent ([ORG]CELL-TP); the code steps are run by the Chief of Staff from the code project
Purpose	decide what is the child's own evidence and declare every crossing
Authority limits	alone: decide membership and adjudicate candidates, with reasons; never: see raw identities or sibling descriptions; create, retype or paraphrase evidence; resolve a contradiction; write a verification label
Write location	the packaging ledger
Reports to	the Chief of Staff
Escalation	to the Chief of Staff in one line; a stop goes to the stop door
Tools and access	none beyond its inputs; no file, search or web tool; no credential in any instruction
Deputy and cover	none; on failure the step halts
Fixed test and score	TEST-TOM-TP v5 (TEST-TOM v5, cases 17, 41, 45)
Card version and status	v0.4.0 · draft

Fixed test

TEST-TOM-TEP v5: TEST-TOM v5, cases 17, 41 and 45 · result NOT_RUN

5.5 Entity Model Design Definition

Field	Content
Name	Entity Model Design
What it is	For one child package, the design of an **architecture-level target operating model**: structure and capabilities, value streams, decision rights and governance skeleton, and expected interface points. A **design rationale and evidence brief** is written at the same time and says why. Each child's design is written by two authors, blind, and reconciled (Annex G).
Purpose	Give the child an operating model that is honest about its evidence and stops where its evidence stops.
Scope	Does: design inside one child; state the entity type it used; record rejected alternatives; mark candidate finer structures as non-authoritative. Does not: design across children (Hand-offs pairs their crossings); write job descriptions, processes, conventions, standards or tool choices; assign work to anything other than an accountable role; size teams; set success criteria; discover the present state beyond the package.
Inputs	One child package (§5.4), and the reference-model catalogue at the run's pinned version (a method, never evidence; Annex B.5); nothing else. Its machine-readable form is the contract of record.

Field	Content
Outputs	**Output A, the TOM Blueprint:** A0 framing; A1 entity type used, with basis and confidence; A2 capability map, with each capability's delivery model, make-or-buy and maturity (Annex H.2, H.4); A3 value streams at capability-level phases, each critical or not, with its reasons (Annex H.3); A4 decision rights and governance skeleton (domains, accountable role-types, interfaces; never policy text); A5 expected interface map, in two separated parts, **stated interface points** (authoritative) and **inferred or unresolved interface candidates** (not authoritative); A6 run status. **Output B, the Rationale and Evidence Brief:** B1 why this structure; B2 evidence ledger; B3 classification basis and what would flip it; B4 alternatives rejected; B5 non-authoritative candidate structures; B6 assumptions, unknowns and limitations; B7 demand signals, by type only; B8 revision record, with no verdicts; B9 what future evidence would change the design. Both outputs are rendered from one **design ledger**, the machine-readable record later stages read (docs/OUTPUT.md §6). The two locked drafts and the difference register (Annex E.18) are run records.
Done when	Gate T5, per child (§5.0).

Field	Content
Limits	The twelve rules of docs/CARRIED-RULES.md §8, in short. One child only; architecture level only; no success criteria. Rationale is captured with the design, never afterwards, with at least one genuine rejected alternative. Every element is traced to the brief, and the trace is checked for fidelity, not presence. Classification is stated with its confidence. Content is data. At most 8 rounds. **The Blueprint stops at the coarsest grain the child's own evidence can tell apart from equally reasonable alternatives.** A finer structure may appear only in the brief, labelled NON-AUTHORITATIVE CANDIDATE — not part of the Blueprint. **Register discipline:** each interface entry carries its crossing reference, marker and direction, and the register's provenance and confidence, never upgraded (a downgrade needs a recorded reason). The Designer never adds a crossing. A stream that needs an undeclared crossing is marked interface-dependent and raised as an undeclared-crossing finding, which returns the package to Entity Packaging. An unresolved crossing is never resolved by assumption. Nothing is designed on the far side. **Narrative cells explain and never add.** A cell names nothing absent from its linked ledger entries, cites its basis entries, and is omitted when it cannot be written without new design content. **What stated means:** a ledger element is stated only where an atom labelled source-backed or organisation-stated states it directly. An element resting on an atom capped at inferred is at most inferred. The child's ceiling is the weakest Annex B label among the atoms behind its load-bearing elements, or assumed where a load-bearing element rests on no atom (Annex B.5). The grain follows the run depth (Annex F): the evidence may stop it earlier, never later. **New in v0.6.0 (Annex H.2–H.4).** A delivery model names how a capability is delivered, never by whom. A maturity level now is only what a held source states; a target only what a source states or the Requesting Authority decides. A stream is critical only on a cited reason.
Sources	49GAION Standard v1.5.0 §3.2 items 2 and 3. Design basis: docs/DESIGN-BASIS.md §5; rule texts: docs/CARRIED-RULES.md §8
Status	draft

New: organisation-stated evidence and reference models.

- An approved statement of the organisation held in Source Lifecycle (an explainer, a confirmation's correction, or a decision's correction) is admitted as evidence with the label organisation-stated (Annex B), statement by statement, keeping each statement's Source Lifecycle label. It carries its label into the ceiling and never becomes source-backed. In the case study, the organisation's statements entered through conversation; here they enter only as approved, versioned sources.
- **Reference models (New in v0.4.0).** The Designer and the Second Designer may read the reference-model catalogue of the 49GAION Library, at the version and hash pinned for the run (contracts/PINS.json; the run preview). A pattern is a method, never evidence (Annex B.5):

  1. It may suggest a structure: capabilities, decision rights by role-type, hand-offs. It never supplies a fact about the organisation, never enters the Input Pack, and never raises a label.
  2. Each ledger element a pattern informed records it in pattern_refs: the pattern, the catalogue version, and what it suggested. B3 states each pattern considered, why it fits and what would flip it (CARRIED-RULES.md §8.1 rule 7).
  3. Each element still rests on the child's own atoms. An element whose basis holds no atom is assumed, with what breaks if it is false and when to re-check it; it is listed in B6 and walked as an assumption. A major or core capability on a pattern alone is never authoritative: it stays at the coarser grain, with the pattern's structure as a non-authoritative candidate.
  4. Only patterns of the pinned catalogue are cited. While the Library holds no catalogue, or none is pinned, the catalogue is declared unavailable and no pattern is cited; none is invented.
  5. The release names each pattern used, in plain words, with the elements that rest on it alone (the Decision Package; the honesty ceiling; the quality checks).
  6. **Done when:** the design check finds every pattern reference in the pinned catalogue and every element on no atom assumed (CARRIED-RULES.md §8.3 item 9); RP5 finds no pattern used as evidence; T9 re-checks every reference by version and hash.

Process (SOP)

Field	Content
Trigger	A child package passes gate T4.
Steps	see the numbered steps below the table
States	designing → locked → revised → checked → reviewed → complete · or returned to packaging (undeclared crossing) · or incomplete-design (naming the blocking element and the missing fact-type) · or held.
Failure and recovery	A material design choice unresolved at the round cap makes the job incomplete-design, never an authoritative Blueprint. The same holds when the coarsest defensible grain is not enough for the model to serve its purpose. The job then names the element and the specific fact-type that would resolve it, and that becomes a needs request. A job may stay complete at a coarse grain only with a stated reason naming which later decision it does not block. A child that ends incomplete-design is put to the Requesting Authority as a WITHDRAW choice (§5.8), so the rest of the slice can go on. Two failures on the same step stop the step and go to the TOM Design Lead.
Records	The ledger versions, the rendered outputs, the revision record, the gate report and the RP5 record.
Cadence	one job per child
Measures	Elements by provenance; interface points stated, inferred and unresolved; alternatives; candidates; uncited elements flagged (must end at 0).

1. **Chief of Staff (runs the receipt code step)**: validates the package against its schema; a mismatch goes back to packaging → receipt (evidence: mechanically-verified).
2. **Designer and Second Designer**: each writes a design ledger for the child, blind to the other (Annex G.2). Each element has its type, claim, provenance (stated, inferred, unresolved, or assumed where its basis holds no atom), confidence, basis references and class (authoritative, candidate, narrative), and, where a reference model informed it, its pattern references. A

decision-domain element names its decision and its accountable role-type. A capability carries its delivery model, the crossings that deliver it where it is not in-house, and its maturity now and target; a value stream carries critical, with its reasons (Annex H.2–H.4). The ledger also holds the narrative cells, bound to the entries they draw on; the alternatives; the candidates; and the demand signals → two ledger drafts (evidence: run record).

3. **Chief of Staff (runs the render code step)**: validates each ledger against its schema, renders Outputs A and B from each, and locks both drafts by hash → two rendered drafts, locked (evidence: hashes).

3a. **Chief of Staff (runs the compare code step)**: aligns the two locked ledgers and writes the difference register (Annex G.4; Annex E.18) → difference register (evidence: hashes; mechanically-verified).

4. **Ledger Reviser**: works in a fresh context, with only the package, the two locked drafts, the difference register and the rules. It first reconciles the two drafts into one ledger under Annex G.5, recording each disposition; then it revises that ledger as before. Its record states only what changed, and each disposition with its reason and quotes, with no verdict → reconciled and revised ledger (evidence: run record). With one author (Annex G.7), steps 3a and the reconciliation are skipped, and it revises the one locked draft.

5. **Chief of Staff (runs the design-check code step)**: renders Outputs A and B from the revised ledger and runs the code checks of docs/CARRIED-RULES.md §8.3. Register to interface: one-to-one. Externals: referenced only by marker. Provenance, confidence and direction: none upgraded from the register. Candidate names: none in Output A outside A5's non-authoritative part, swept on every narrative cell and again on the rendered Output A. Narrative cells: no negative enumeration; no confidence or provenance vocabulary; no name absent from the cell's linked entries. Every element: has a brief entry. Schema: no undeclared section. Patterns: every pattern reference resolves in the pinned catalogue; an element on no atom is assumed. Role fields: every decision domain names its decision and its role-type, or unknown. Delivery and maturity: every capability's delivery model is in the closed list, with the make-or-buy that follows from it and its crossings where it is not in-house; every critical stream has a cited reason; every level cites its basis (Annex H.11 checks 2 and 3; CARRIED-RULES.md §8.3 items 11 and 12) → gate report (evidence: mechanically-verified).

6. **Review Pod**: review point RP5 (§5.8). It checks fidelity, not presence: every Blueprint element, and every evidence entry supporting one, says exactly what its cited atoms say. Coverage is full; a declared sample is allowed only for entries that support no element. It also looks for semantic paraphrase of candidates. It also checks each difference's disposition (Annex G.6), and that no pattern is used as evidence (Annex B.5) → RP5 record (evidence: externally-audited).
7. **Chief of Staff**: on an upheld or ruled MATERIAL finding, starts bounded repair (§5.8 rule 19); otherwise records the step complete → step state (evidence: run record).

Role card

Field	Designer	Second Designer	Ledger Reviser
Name	Designer	Second Designer	Ledger Reviser
Position and seat	Designer; accountable seat the TOM Design Lead	Second Designer; same	Ledger Reviser; same
Performed by	an agent ([ORG]CELL-TD)	an agent ([ORG]CELL-TSD), on an engine of the other author family	an agent ([ORG]CELL-TLR), in a fresh context, never the designer's context
Purpose	design the child and capture why, together	design the child a second time, blind to the Designer's ledger	reconcile the two locked drafts under Annex G.5, then revise the result against the package and the rules
Authority limits	alone: write the ledger; never: see beyond the package; add a crossing; design the far side; write a verdict or a verification label; see the other author's ledger; use a reference model as evidence	alone: write the ledger; never: see beyond the package; add a crossing; design the far side; write a verdict or a verification label; see the other author's ledger; use a reference model as evidence	alone: reconcile and revise the ledger; never: add evidence; add what neither draft holds; choose by preference; raise a provenance or confidence; write a verdict
Write location	the design ledger	the design ledger	the revised ledger and the revision record
Reports to	the Chief of Staff	same	same
Escalation	to the Chief of Staff in one line; a stop goes to the stop door	same	same
Tools and access	none beyond its inputs; no file, search or web tool; no credential in any instruction	same	same
Deputy and cover	none; on failure the step halts	same	same

Field	Designer	Second Designer	Ledger Reviser
Fixed test and score	TEST-TOM-TD v5 (cases 18, 19, 38, 39, 91, 92, 97, 112–115)	TEST-TOM-TSD v5 (cases 18, 19, 38, 39, 91, 92, 97, 112–115)	TEST-TOM-TLR v5 (cases 18, 93)
Card version and status	v0.4.0 · draft	v0.4.0 · draft	v0.4.0 · draft

Fixed test

TEST-TOM-TEM v5: TEST-TOM v5, cases 18, 19, 38, 39, 91-93, 96, 97 and 112-115 · result NOT_RUN

5.6 Hand-offs

Definition

Field	Content
Name	Hand-offs
What it is	**Hand-offs** (short for "hand-offs between parts and with outside parties"). For the decided slice, the work that must pass between its children, and between each child and its outside parties, and nothing above or beside them. It pairs every declared crossing with its counterpart, including crossings between children of one entity. It reports every seam that does not pair and routes it back, never bridging it. For each clean pair it records what crosses, each side's obligation and trigger, the accountable role at each end, and the decision on the crossing, as one field. It keeps the counterparty register by class, for every shape of organisation, and, for each of its entries, the journey of that outside party through the organisation (Annex H.5). It lists the legal and regulatory perimeter for human legal review. It keeps the register of every boundary class not yet designed. It designs no capability, value stream, decision domain or body of the whole: there is no model of the whole beyond the children and the hand-offs between them. A shared service is a hand-off like any other. **New in v0.4.0.** By code, it also lists every accountable role-type of the release in the **role register** (Annex E.17): what each decides, what it is accountable for, where it appears, and the hand-offs it sends and receives. A conflict between roles is a role finding, routed back or put as a decision row, never merged. The register designs nothing: it lists what the children's decided models already hold.
Purpose	Make every hand-off between children, and every crossing with an outside party, visible and owned at both ends, without hiding a seam and without designing anything above the children.

Field	Content
Scope	Does: restore identities inside the hand-offs workspace only; pair crossings; check seams; record each clean pair in the hand-offs register; map counterparties by class; record each outside party's journey; list the perimeter; keep the deferred-boundary register; pair carried children with new ones (§5.9); compile the role register by code and route its findings. Does not: change a child's own model; design a capability, value stream, decision domain or body of the whole; make a legal determination; bridge or reconcile a seam by its own choice; merge two role names; give a decision to a role; settle a role finding by its own choice.
Inputs	The decided design ledgers of the slice, and those of any carried children; their declared boundary registers; the marker-to-identity maps; the withheld-context records; the party register; the deferred-boundary register; the decided Design Brief and structure.
Outputs	The seam findings, the hand-offs register, the counterparty register, the legal and regulatory perimeter and the deferred-boundary register, the role register, and the outside-party journeys (Annex E.4–E.7a, E.17, H.5; docs/OUTPUT.md §7); the pairing table (run record; Annex E.6a).
Done when	Gate T6 (§5.0).

Field	Content
Limits	A crossing with no counterpart, or a pair whose direction, content or obligations disagree, is a **seam finding**. It is routed back as a decomposition (Structuring), packaging or evidence defect, or put as a decision row. It is never bridged, and never reconciled by the Hand-off Analyst's own choice: reconciliation is done by routing back, and Hand-offs never edits a child's model. A child's own accountability stays in its model; the relationship with the counterparty is recorded here or deferred. A relationship between independent entities is named as a relationship, never as a body above them. A shared service is a hand-off like any other. A boundary class is never dropped. Each deferred class carries a durable ID, its evidence by atom, the run that surfaced it, the reason for deferral and its destination. It leaves the register only when a designed capability incorporates it, recorded against its ID. The known deferred classes this capability must be able to accept and place are listed in docs/CARRIED-RULES.md §9.2. The role register is never a model of the whole, and never names a person, a holder, a number of people or a position.
Sources	Standard v1.5.0 §1.7; design basis docs/DESIGN-BASIS.md §6; rule texts docs/CARRIED-RULES.md §9
Status	draft

Process (SOP)

Field	Content
Trigger	Every child of the decided slice is complete, or withdrawn by decision, or carries only MATERIAL findings accepted as known defects (§5.8).
Steps	see the numbered steps below the table
States	pairing → checked → reviewed → complete · or held on an open seam or role finding.
Failure and recovery	A seam finding goes back to the Structure, Package or evidence of the children concerned, or becomes a decision row; never bridged. For a carried child, §5.9 rule 5 applies. Two failures on the same step stop the step and go to the TOM Design Lead.
Records	The pairing table, the seam findings, the registers, the perimeter.
Cadence	per release
Measures	Crossings paired; seams unmatched and contradictory; hand-offs recorded; counterparty classes; perimeter items; deferred classes; roles; role findings by kind and how each was settled.

1. **Chief of Staff (runs the pairing code step)**: restores identities inside the hand-offs workspace only, and pairs each outbound marker with its inbound counterpart, carried children included → pairing table (evidence: hashes).
2. **Chief of Staff (runs the seam-check code step)**: every unmatched or contradictory crossing between children becomes a seam finding (Annex E.6) → seam findings (evidence: mechanically-verified).
3. **Hand-off Analyst**: records each clean pair in the hand-offs register (Annex E.4), from both sides' registers and ledgers, citing sources or decided items only; the decision on the crossing is one field → hand-offs register (evidence: run record).

4. **Hand-off Analyst**: maps every counterparty by class (Annex E.5); raises a need where the sources are silent on a class the organisation's shape normally has → counterparty register (evidence: run record).

4a. **Hand-off Analyst**: records the journey of each counterparty entry: its moments in order, each a crossing of that entry, with the stream and phase it meets; a moment no crossing carries becomes a seam finding (Annex H.5) → outside-party journeys (evidence: run record). 4b. **Chief of Staff (runs the journey and delivery code step)**: checks every journey moment against its entry and every phase against the decided ledgers, and every crossing that delivers a shared service, outsourced or partner capability against the hand-offs register, the counterparty register and the deferred-boundary register; each miss is a seam finding (Annex H.11 checks 2 and 4) → check report (evidence: mechanically-verified).

5. **Hand-off Analyst**: lists the legal and regulatory perimeter, each item flagged for human legal review → perimeter list.

6. **Hand-off Analyst**: places or registers every deferred boundary class (Annex E.7) → deferred-boundary register.

6a. **Chief of Staff (runs the role-register code step)**: compiles the role register from the decided ledgers of the slice and of carried children, the hand-offs and counterparty registers and the obligations of the decided Design Brief, by the rules of Annex E.17; raises the role findings; sends each decision row finding as a non-delegable row (§5.2) and routes the others → role register and role findings (evidence: hashes; mechanically-verified).

7. **Review Pod**: review point RP6 (§5.8) → RP6 record (evidence: externally-audited).

Role card

Field	Content
Name	Hand-off Analyst
Position and seat	Hand-off Analyst; accountable seat the TOM Design Lead
Performed by	an agent ([ORG]CELL-THA)
Purpose	record each hand-off honestly at both ends, and map every counterparty
Authority limits	alone: record clean pairs; map; register; never: change a child's model; bridge or reconcile a seam; decide a legal question; design anything of the whole
Write location	the hand-offs records
Reports to	the Chief of Staff
Escalation	to the Chief of Staff in one line; a stop goes to the stop door
Tools and access	none beyond its inputs; no file, search or web tool; no credential in any instruction
Deputy and cover	none; on failure the step halts
Fixed test and score	TEST-TOM-THA v5 (cases 20, 37, 42, 65, 66, 70, 87, 116)
Card version and status	v0.4.0 · draft

Fixed test

TEST-TOM-TH v5: cases 20, 37, 42, 65, 66, 70, 85-88, 113 and 116 · result NOT_RUN

5.7 Operational Walkthrough

Definition

Field	Content
Name	Operational Walkthrough
What it is	A structural walk of the operating model through realistic situations. It finds whether the model's responsibilities, authorities, decisions, boundaries and dependencies are **navigable, closed, and honest about where they stop**. In plain words: when a real situation arrives, can you find your way through this operating model, and does it tell you honestly where it stops? It is **not a simulation**. It makes no claim about throughput, staffing, time, cost or volume.
Purpose	Stop a model that has a material defect on a path it claims to handle, before anyone decides it.
Scope	Does: compile the model to a graph; walk sealed situations deterministically; classify findings; report. Does not: redesign; propose wording; fill an unknown; invent a role, team, title or reporting line; play out the far side of a boundary; decide anything.
Inputs	The **walk input** (Annex E.8), compiled by code from the decided Design Brief, the design ledgers, the hand-offs and counterparty registers and the deferred-boundary register: six sections, each citing its part and hash. The **blind brief** (Annex E.9), for the Situation Author. The situation library version, and the situation pack of the request kind. The run depth (Annex F), which sets the situation budget above its floors. The six sections are read as: established facts (may be relied on); the model (relied on as designed, never as observed); labelled assumptions (usable, with the flag carried); unknowns (may never be filled); deferred or excluded scope (a legitimate halt); outside dependencies (declared crossings and their conditions).
Outputs	The **walkthrough report** (docs/OUTPUT.md §8), readable by the Requesting Authority, with the finding register, the trace, the coverage and the termination profile.

Field	Content
Done when	Gate T7 (§5.0).
Limits	Situations are hash-sealed before the walk; changing one afterwards voids the walk. The seat that designed the model never authors the situations. An engine may only propose the next hop from a node's own text; a hop counts only if the binding rules license it (docs/CARRIED-RULES.md §10.1), and no finding rests on free role-play. The same model, the same situations and the same rules give the same report. No numeric performance claim, ever. A clean report reads "no defect found by an instrument of partly measured sensitivity", never "no defect exists". Situations may inject outside events of a class the entity's type supports, even where the model is silent. Two authorities claiming one decision are not a defect where joint ownership is declared. A field the model does not declare is a finding, never a compile failure. In a mandatory negative, invoking an excluded, deferred or withdrawn element is a finding, unless the expected outcome is a declared halt or routing to deferred scope. A reduced-catalogue walk says "reduced: value streams not walked" in the report's headline and coverage.
Sources	49GAION Standard v1.5.0 §3.2 items 2 and 3. Design basis: docs/DESIGN-BASIS.md §7; rule texts: docs/CARRIED-RULES.md §10
Status	draft

Situation types.

- **W1:** the happy path of each value stream.
- **W2:** an in-scope exception, or a demand to discharge a held obligation.
- **W3:** authority ambiguity, a gap, or a decision that cannot be taken locally.
- **W4:** an outside dependency that fails, or whose condition is not met.
- **W5:** an assumption flipped false, a critical-path unknown made unavailable, or an excluded class probed.
- **W6:** two live tokens competing for one decision domain (narrowed, below).

- **Mandatory negatives:** for every excluded class, deferred class and withdrawn candidate, a situation that must not need it.

Situation sources.

- At least 60% from the standard situation library, written against finding classes before any particular model exists; the request kind's situation pack selects from it.
- At least 30% from a blind adversarial pack, written from the blind brief only (Annex E.9): the entity's type, accountability set and boundary list, never its capability map, streams or domains.
- The rest named by the organisation: additive only, and never able to rescue a core failure.
- **New.** Until the library holds at least one situation per finding class for the entity's type, the walk runs with the shortfall declared in the report. The mix rule is then reported as not met, never silently waived. The library is a build item of the code project (§8).
- **New in v0.4.0.** The budget follows the run depth (Annex F.2). The floors above hold at every depth; the situations deep adds are additive.

Finding classes.

- CLEAR: closed in a licensed terminal state, on facts and design only.
- LIMITATION: closed or halted honestly. The cause is a flagged assumption, an unmet condition with a declared holding obligation, an excluded or deferred halt, or an unknown on a path the model never claimed.
- MINOR: a wording or traceability defect that does not change navigability.
- MATERIAL: a defect on a path the model **claims** to handle. It is one of:
  - unowned work;
  - an unarbitrated collision;
  - an authority violation with no escalation home;
  - an undeclared boundary escape;
  - a scope leak;
  - a withdrawn candidate brought back;

  - a loop with no exit;
  - a critical-path unknown;
  - responsibility that evaporates with no holding obligation;
  - a claimed obligation with no way to discharge it.
- NOT-WALKABLE: the model cannot compile, or the situation needs evidence the input marks absent.

Each finding states what was injected, where the path stopped, which element and which class, and what information or decision would close it. **It names the gap, never the fix.** The licensed terminal states and the per-hop and per-walk fields are in docs/CARRIED-RULES.md §10.

New: the five known defects of the earlier standard are fixed (docs/CARRIED-RULES.md §10.4):

- I-1: a halt at excluded scope has a licensed terminal state, halts-excluded-scope.
- I-2: a standing evidence ceiling is declared once, as the frame of the walk, not as a flag on every path.
- I-3: the class follows the claimed-path test, and the terminal follows the licensed-terminal list.
- I-4: W6 detects a defect only where the two decisions are mutually exclusive and held by different authorities.
- I-5: a finding's class must cite the cell text that licenses it; a class resting on an unbindable binding is void.

The reduced-catalogue rule is adopted: a model complete in obligations and authority but thin in value streams is walked on its capability, domain and boundary graph. It is never refused as not walkable for being thin, and its grade is "PASS WITH LIMITATIONS" or "SIMULATION-INCOMPLETE", never a plain pass.

Process (SOP)

Field	Content
Trigger	The hand-offs are complete, or carry only MATERIAL findings accepted as known defects (§5.8).
Steps	see the numbered steps below the table
States	compiling → sealed → walking → classified → reviewed → reported.
Failure and recovery	A MATERIAL finding stops the release. The Chief of Staff sends the Requesting Authority a **material finding notice** as one decision row (§5.8). It states the finding and what is missing (the gap, never the fix), and offers the correction choices with their costs and one recommended path; none is adopted until the Requesting Authority decides. Before any correction runs, a closure test with its pass mark is written and sealed. A statement of the organisation admitted as evidence never closes a finding by itself. The finding closes only when the corrected model passes the sealed closure test, and the original failing situations, re-run as regression, change outcome. A fresh situation set never erases a recorded failure. Budget exhaustion returns an incomplete walk, never a silent cut.
Records	The walk input, the blind brief, the seal record, the trace, the finding register, the RP7 record, the closure tests and the report.
Cadence	per release
Measures	Coverage, with named gaps. Findings by class. Hop count and escalation depth. The termination profile: closed, halted at a boundary, halted at deferred or excluded scope, halted on an unknown, halted on unbindable text. Ratings only on executability, accountability clarity, authority coherence and boundary operability, where the evidence supports them; everything else "not assessed".

0. **Chief of Staff (runs the walk-input code step)**: compiles the walk input and extracts the blind brief → walk input, blind brief (evidence: hashes).

1. **Chief of Staff (runs the compile code step)**: compiles the model and the hand-offs to the obligation, authority and boundary graph. A cell that cannot be bound is a compile finding (unbindable-prose), never a guessed edge → graph and compile findings (evidence: hashes).
2. **Situation Author** (a seat that did not design the model): writes the blind pack from the blind brief only. The code generates the standard and negative situations from the situation pack of the request kind, and the budget is declared before the walk → situation set (evidence: run record).
3. **Chief of Staff (runs the seal code step)**: hash-seals the model, the situation set and the rules version → seal record (evidence: hashes).
4. **Situation Walker**: walks each situation hop by hop under the binding rules. For each hop it records: the input; the actor; the element; the decision; the authority relied on; the licensing rule; the crossing; the assumption flags; the outcome; the terminal state → trace (evidence: run record).
5. **Situation Walker**: classifies each finding, citing the cell text that licenses the class → finding register (evidence: run record).
6. **Review Pod**: review point RP7 (§5.8). It reviews the findings only, never re-walks and never redesigns. It confirms by hash that the walked model is the reviewed model, and replays one or two traces → RP7 record (evidence: externally-audited).
7. **Chief of Staff (runs the render code step)**: renders the report → walkthrough report (evidence: hashes).

Role card

Field	Situation Author	Situation Walker
Name	Situation Author	Situation Walker
Position and seat	Situation Author; accountable seat the TOM Design Lead	Situation Walker; same
Performed by	an agent ([ORG]CELL-TSA), never the designer's context	an agent ([ORG]CELL-TSW), with the deterministic walk of the code project
Purpose	write situations blind to the design	walk and classify under the rules
Authority limits	alone: write the blind pack; never: see the capability map, streams or domains	alone: walk and classify; never: fill an unknown; invent an actor; propose a fix
Write location	the situation set	the trace and the finding register
Reports to	the Chief of Staff	same
Escalation	to the Chief of Staff in one line; a stop goes to the stop door	same
Tools and access	none beyond its inputs; no file, search or web tool; no credential in any instruction	same
Deputy and cover	none; on failure the step halts	same
Fixed test and score	TEST-TOM-TSA v5 (cases 21, 84, 91)	TEST-TOM-TSW v5 (cases 21, 71)
Card version and status	v0.4.0 · draft	v0.4.0 · draft

Fixed test

TEST-TOM-TW v5: TEST-TOM v5, cases 21, 71, 84 and 91, and the walkthrough battery · result NOT_RUN. The battery is a build item. It sets the card's claim and the wording of a clean walk; no walk or release waits for it (tests/TEST-TOM.md).

5.8 Independent Review

In plain words. At the end of every stage, three or more AI engines check the stage's output. They come from at least two families. In normal mode none is from the family of the engines that made the output. Each engine first reviews alone. Then each sees the findings of the others, never who raised them or how many did, and confirms, challenges with a quote, or says it cannot check. They do not vote. Code checks every quote and settles every finding. A human, the TOM Design Lead, rules on what stays contested. This is the **Review Panel** rule; a Review Panel is the set of engines drawn for one review. It is not the Panel of §4.4.

Definition

Field	Content
Name	Independent Review
What it is	The rule that closes every stage. At each review point (table below), a Review Panel of engines, drawn from the engine register at the tool gateway into the seats of the Review Pod [ORG]POD-TR, examines one locked output. First each engine reviews alone (the blind pass); then the engines exchange positions on the findings only (the debate). Code checks every quote, settles every finding and computes the verdict.
Purpose	A step's own checks share its blind spots, and one reviewer shares its own. Several engines from several families, held to quotes and settled by code, keep every disagreement visible instead of averaging it away (Standard §2.3 item 9; §2.4 item 1).
Scope	Does: the packet, the draw, the blind pass, the quote and coverage checks, the debate, the settlement, the verdict, the review record, the ruling sheets, bounded repair and re-review. Does not: redesign; propose wording or repair; vote; rule; review its own work; use a tool; put anything about itself into a release part, except the quality checks made by code (Annex E.11).
Inputs	One **review packet** per review point, all by hash: the stage's inputs; its locked output; the frozen rules; the review point's questions; the Input Pack version the stage read. Nothing of the dialogue that made the output, except at RP1b, whose object is that dialogue. No author identity. On a re-review, no earlier finding.
Outputs	One **review record** (a run record, never a release part; Annex E.10). A **verdict** to the Chief of Staff: CLEAR, MINOR-findings-only or MATERIAL-findings-present, with the level full or reduced. **Ruling sheets** for contested MATERIAL findings.

Field	Content
Done when	The review record validates. Every blind card was complete before any round-1 call began. Every quote passed its check, or its finding is void. Every checkable item was examined by at least two engines, or is recorded not covered. Every finding is upheld, withdrawn, void or contested. Every contested MATERIAL finding is ruled. A second run of the settlement step gives the same verdict. A review so done is **settled**. It is **clear** when its settled verdict is CLEAR or MINOR-findings-only.
Limits	Rules 1–20 below.
Sources	49GAION Standard v1.5.0 §2.3 item 9 (no unit grades its own work), §2.4 item 1 (shared blind spots), §3.2 item 2, §4.4 items 4, 8 and 11; Human Floor HF-18; Operating Protocol v1.5.0 §10 item 6. Design basis: docs/DESIGN-BASIS.md §8; rule texts: docs/CARRIED-RULES.md §11.
Status	draft

Rules

1. **Where it applies.** At every review point of the table below, once per locked output, and per child at RP3, RP4 and RP5. A stage's gate passes only on its settled review.
2. **Engines.** An **engine** is one AI model as served through the tool gateway. Its **family** is the lineage the engine register records for it (Annex E.12). A Review Panel has as many engines as the slot review_engines sets: at least 3, at most 5 (§15). Two engines of one family count once wherever families are counted. At the run depth deep, every Review Panel of the run has five engines, and the draw takes three families where the register allows (Annex F).
3. **Families and the author.** A Review Panel holds engines of at least as many families as the slot review_families_min sets (at least 2), and three where the register allows. The **author** of an output is every AI Cell whose engine produced it, as the gateway log shows. For RP0 and RP9, whose outputs are made by code steps, the author is the Chief of Staff, which runs them. An output with two authors (Annex G) has both as authors, and its reconciler,

whose engine is of one of their two families; so it has exactly two author families. In normal mode no engine of an author's family sits on the review.

4. **Level.** full: at least three engines, at least two families, none of an author's family. reduced: an author's family sits on the review, or all engines are of one family. With fewer than three eligible engines no review takes place: the step stays produced, not reviewed, and the run shows held: no reviewers (Annex C.1).
5. **Blind pass.** Every engine receives the same packet in its own fresh context, with no tool and no memory, and sees nothing of any other engine. The order of the questions, and of packet sections that have no order, is shuffled for each engine. Each engine returns one **card**: a verdict; its findings; the items it examined; the items it could not check, with the reason. The calls run in parallel. No round-1 call starts until every blind call has returned or failed. Each seat calls under its own credential (§17).
6. **A finding** holds: a class (MATERIAL or MINOR, as the review point defines them); a target (the ID of an item of the output); quotes of the output and of the input or rule; the mechanism, in at most 40 words; and what would close it (the gap, never the fix). An engine never proposes wording, design or repair.
7. **Quotes.** Engines see the source text in their packet. Every quote in a card, a position or a record is held as atom ID (or packet item ID), span and span hash, never as text (docs/CARRIED-RULES.md §11.2). Code checks every quote against the packet. A quote that fails goes back to its engine once; if it fails again, its finding is void. Free text (a mechanism or a position) holding eight or more consecutive words of a packet source is sent back once; if it fails again, the position counts as CANNOT-CHECK.
8. **Coverage.** Every checkable item of the output (an element, a crossing, a numbered statement, a row) is examined by at least two engines. Code sends items examined by fewer to the engines that examined fewest, for one targeted pass. An item still short is recorded not covered, and code raises a MINOR finding on it, so the verdict cannot be CLEAR.
9. **No debate when clear.** When every blind card is CLEAR, no finding was raised and every item is covered, the review settles CLEAR without debate.

10. **Round 1: cross-examination.** Code merges findings with the same target and class, removes every sign of who raised them and how many did, sets each in a fixed form, and shuffles their order for each engine. Every engine takes one position on every finding:
    - CONFIRM: it checked, agrees, and gives its own quotes;
    - RECLASS: a defect exists, the class differs; it quotes the rule that fixes the class;
    - CHALLENGE: there is no defect; it quotes the text that shows it;
    - CANNOT-CHECK: with the reason.

A position is at most 120 words. An engine may add at most three new findings, marked late, with the same quotes. Silence is not a position: a missing position is recorded CANNOT-CHECK.

11. **Round 2: rebuttal.** Only for findings challenged or reclassified in round 1, and for late findings. The raiser sees each quoted challenge; each challenger sees the raiser's mechanism. Each states a final position, HOLD, WITHDRAW or CONCEDE, with a packet quote. "The others disagree" is not a reason.
12. **Stop.** The debate stops at the first of: every finding settled; a round in which no position moved and no finding was added; two rounds (the cap, fixed by this rule); the review's budget (rule 16). No round is added to reach agreement.
13. **Settlement, by code, without a vote.** Each finding ends as one of:
    - upheld: after the last round no CHALLENGE or RECLASS stands on it, and at least one engine of a family other than the raiser's has confirmed it with its own quotes;
    - withdrawn: its raiser withdrew it, with a quote;
    - void: its quote failed twice (rule 7);
    - contested: anything else, including a finding no other family could check.

A finding's class is the highest class any engine still holds. The verdict is the worst class among the findings that end upheld or contested: MATERIAL-findings-present; else MINOR-findings-only; else CLEAR.

14. **Nobody votes a finding away.** A finding with a verified quote leaves the register only when its raiser withdraws it, or when the TOM Design Lead rules it out. Not by the number of engines or families, not by the Chief of Staff, not by code. A MATERIAL finding withdrawn by its raiser is shown on the Panel, and the Design Lead may reinstate it.
15. **Who rules.**
    - An upheld MATERIAL finding goes straight to bounded repair (rule 19), or, past the repair cap, to the material finding notice. The Chief of Staff confirms it and rules nothing.
    - A contested MATERIAL finding goes to the TOM Design Lead on a one-page **ruling sheet**: the finding; the quotes on each side; each family's last position; the rule text; and three choices, UPHOLD, UPHOLD AS MINOR and DISMISS. The last two need a written reason. The ruling rests on the rule text and the evidence, never on a count, and is signed with a live sign-in. Before ruling, the Design Lead may ask for one further engine of a family not yet on this review; its position is evidence, not a vote.
    - A contested MINOR finding holds nothing. It is recorded with its dissent.
    - While a ruling is awaited, the step shows waiting: ruling. Silence rules nothing.
    - At the reduced level no other family can confirm a finding, so every MATERIAL finding is contested and goes to a ruling.
16. **Cost.** Calls per review are at most the number of engines × (1 blind pass + 1 targeted pass + 2 rounds), plus one re-send per returned quote (rule 7) and the one further engine of rule 15. Each run has a review budget (slot review_budget_per_run, set by the Cell carrying [ORG]HOAC-TM, within budget_cap), metered at the gateway. The Chief of Staff tells the Design Lead at 80% of it. At 100% the step stops with held: review budget. Nothing lowers the engines, families, rounds or level to save cost.
17. **Eligibility, ceilings and the served engine.**
    - An engine is eligible for a review only if the engine register lists it as active for reviewing at that review point, its confidentiality ceiling is at or above the highest level of any atom in the packet, and it has passed the reviewer seat test (§9) at its current version.

- No atom reaches an engine above its ceiling (Annex D item 18).
- The identity of the engine actually served is checked on every call. A mismatch voids the call, which is re-run once. After a second mismatch the engine is dropped from that review.
- A packet larger than an engine's context is split by code into groups. The same engines review every group, and coverage (rule 8) proves each item was examined. A packet is never truncated.

18. **Reduced mode.** A reduced review takes place only where the slot reduced_review_permitted (set by the Cell carrying [ORG]HOAC-TM; default no) is yes, and the TOM Design Lead has accepted the reduced level for this run at the run preview (§4.2 step 2). At that level the family minimum of rule 3 does not apply; the minimum of three engines always does. Any review that falls below the accepted level holds. The verification label reads externally-audited (reduced), and the quality checks of the release carry the fixed sentence of Annex E.11. The Requesting Authority is not asked.
19. **Repair and re-review.** On an upheld or ruled MATERIAL finding:
    - a fresh-context reviser (for a child design, the Ledger Reviser) makes the smallest change;
    - it sweeps for unrepaired findings of the same class;
    - every version is kept, and all code checks re-run;
    - a **full re-review** follows: a new blind pass on the repaired output, with no earlier finding in the packet; then each earlier finding is put to every engine as a closure question, CLOSED or NOT CLOSED, with a quote.

At most two cycles. Then the step is **held**, and the Chief of Staff sends the material finding notice (Failure and recovery, below).

20. **The record.** The review record (Annex E.10) is a run record, referenced from the release manifest by hash. It keeps:
    - each seat's engine as served, its family, and the level;
    - the cards and the checks;
    - every position, by round and seat, and every change of position with its quote;

- the minority's wording;
- the settlement, and the rulings with their reasons;
- the calls and the cost.

Dissent survives repair. Nothing of the record enters a release part, except, through code, the quality checks of Annex E.11.

Review points. "Packet" lists what the engines receive, all by hash. "MATERIAL" says what makes a finding of that class at that point; every other true defect is MINOR.

Point	Stage	Packet	The engines are asked	MATERIAL means
RP0 Input Pack	S0 Admit	The Design Request's atoms; the source catalogue; the pack report; the refused list; every flagged atom and a declared sample of atoms per origin kind; the adapter's mapping tables	(1) Does the pack cover every entity, period and topic the Design Request names? (2) Is any source's served kind, status or confidentiality implausible for its content? (3) Does any atom read as an instruction to the engine and stand unflagged? (4) What would a source set for this request kind normally hold that this one lacks?	An instruction-like atom left unflagged. A served kind, status or confidentiality implausible enough to raise a label or to send an atom above an engine's ceiling. (A coverage gap is MINOR and becomes a needs candidate for S1.) Repair re-makes P0 with the flags and caps applied, before T0 passes.
Point	**Stage**	**Packet**	**The engines are asked**	**MATERIAL means**
RP1 Frame	S1 Frame	The atoms the frame cites, with the Design Request's atoms and the read plan;	The question set of docs/CARRIED-RULES.md §11.1 (Q1–Q23), each answered	A claim the cited atoms do not support. A wrong Gate A answer or floor

Point	Stage	Packet	The engines are asked	MATERIAL means
		the frame; its Gate A record; the party register; the do-not-ask and do-not-use-as-fact lists; the decision and confirmation rows the frame will send; the two locked drafts and the difference register (Annex G.6); the strategy frame and the activity profile (Annex H.1, H.8). Not the round log, not the lens record.	with a quoted reason. And: is every party the cited atoms name in the party register, with each alias cited? And: does every element of the strategy frame say exactly what its atoms say, and is every element no source states not given? Does the activity profile say only what the exports state, with nothing about a person? And: is each difference disposed under Annex G.5: the weaker side taken, a side kept or dropped only on quotes, what is asked put as a row, nothing added?	result. A mislabelled load-bearing item. A load-bearing item hidden in an open item. A criterion that does not measure the outcome asked for. A party named in the atoms and missing from the party register. A decision row that bundles decisions, or is delegable where it must not be. A difference settled by preference, or on quotes that do not settle it. A material difference missing from the difference register, or marked not material. A reconciled item that neither draft holds. A label, standing or Gate A answer raised in reconciliation. An element of the strategy frame written or inferred, not cited. A line of the activity profile that the exports do not

Point	Stage	Packet	The engines are asked	MATERIAL means
				state, or that is about a person.
RP1b Process audit	S1 Frame	The round logs; the lens records (of both authors); the question typology (CARRIED-RULES.md §2.3 item 3); the guard targets (§2.1 item 3). This is the only packet that holds the dialogue of the work, because the dialogue is what it audits.	Is there an invented answer, a leading question, label drift, or a forbidden question type? Did a lens finding enter the Working Context without a citation?	An invented answer. A forbidden question that survived. A label that drifted upward. An uncited lens finding in the Working Context. A MATERIAL verdict stops the step, and the frame is re-formed from the last clean round (CARRIED-RULES.md §2.1 item 5).
RP2 Structure	S2 Structure	The decided frame; the Structure Proposal; the orphan table, the exclusion matrix, the four-leg answers, any attempted-split table, the alternatives, the dependency order and the first slice; the party register's extension	Is each of the four legs answered honestly? Is any orphan or excluded item smuggled in? Is design hidden in a child description? Is the set the smallest that passes? Does the first slice bind only as a decision row?	An orphan. An exclusion crossed. Design inside a child description. A leg answered against its evidence. A live dependency cycle.
Point	**Stage**	**Packet**	**The engines are asked**	**MATERIAL means**
RP3 Child brief, per child	S3 Child briefs	As RP1, for the child: its decided	The question set of CARRIED-RUL	As RP1. An inadmissible

Point	Stage	Packet	The engines are asked	MATERIAL means
		structure row; the decided frame; the child brief; its Gate A record; its criteria rows	ES.md §11.1, for the child. Is each criterion admissible (Annex C.3)? Does the floor hold for this child?	criterion counted toward the floor.
RP4 Package, per child	S4 Package	The markerised atoms; the child row; the package; the packaging ledger; the G1–G5 report; the rules (CARRIED-RULES.md §7)	Is far-side content left after substitution? Is there semantic identity leakage? Is an evidenced crossing not declared? Is the child's own evidence withheld?	Far-side content. A semantic identity leak. An undeclared evidenced crossing. Own evidence withheld.
Point	**Stage**	**Packet**	**The engines are asked**	**MATERIAL means**
RP5 Design, per child	S5 Design	The package; the locked ledger; Outputs A and B; the revision record; the design-check report; the rules (CARRIED-RULES.md §8); the two locked drafts; the difference register; the reference-model catalogue at the run's pinned version	Does each element say exactly what its cited atoms say (fidelity, not presence)? Is any structure finer than the evidence? Does a candidate leak into Output A, by name or by paraphrase? Is register discipline kept? Is each difference disposed under Annex G.5? Is any pattern used as evidence: an element stated	An element that adds to, drops or alters what its atoms say. A candidate leak. An upgraded provenance, confidence or direction. A minted crossing. A difference settled by preference, put as a design question, or missing. A reconciled element that neither draft holds. A pattern used as evidence. A

Point	Stage	Packet	The engines are asked	MATERIAL means
			or inferred on a pattern alone, or a major or core capability authoritative on a pattern alone? Does any delivery model name a provider? Does any maturity level, or any reason a stream is critical, rest on less than its cited atoms or decision?	pattern cited that the pinned catalogue does not hold. A provider named in a delivery model. A maturity level assessed by an author. A stream made critical without a cited reason.

Point	Stage	Packet	The engines are asked	MATERIAL means
RP6 Hand-offs	S6 Hand-offs	The child ledgers; their boundary registers; the pairing table; the seam findings; the hand-offs register; the counterparty register; the perimeter list; the deferred-boundary register; the role register and its role findings; the outside-party journeys and the journey and delivery check report (Annex H.5, H.11); the rules (CARRIED-RULES.md §9)	Is any seam bridged by choice? Does any journey add a crossing, or design anything on the party's side? Is any hand-off field (what crosses, each side's obligation and trigger, the role at each end, the decision on the crossing) unsupported by both sides' evidence or a decision? Is anything designed for the whole (a capability, stream, domain or body above	A bridged seam. An unsupported hand-off field. Design of the whole. A legal determination. A dropped boundary class. A role entry that adds or drops a decision, an accountability or a hand-off. Two names joined without a same role decision. A role finding missing or settled by choice. A person named. A journey that adds a crossing or designs the party's side.

Point	Stage	Packet	The engines are asked	MATERIAL means
			the children)? Is any legal question decided? Is any boundary class dropped? Does each role entry say exactly what its ledgers and registers hold? Is any role merged, renamed or given a decision without a decision row? Is a conflict missing from the role findings, or routed by choice? Does the register add any role or body above the children?	
RP7 Walk	S7 Walk	The graph hash; the walk input; the sealed situation set; the finding register; one or two traces chosen by code; the rules (CARRIED-RULES. md §10). Never the Situation Author's dialogue.	Does each class match the rules and cite the cell text that licenses it? Does any finding rest on role-play? Is the walked model the reviewed model, by hash? Does a replayed trace reproduce?	A class not licensed by cell text. A finding on role-play. A walked model that differs from the reviewed one. A trace that does not replay. There is no second walkthrough and no auditor of the auditor.
Point	**Stage**	**Packet**	**The engines are asked**	**MATERIAL means**
RP8 Decision Package	S8 Decide and release	The Decision Package; the	Does every statement trace	A statement beyond its part.

Point	Stage	Packet	The engines are asked	MATERIAL means
		reviewed parts it summarises; the decision rows to be sent; the quality checks	to a reviewed part and claim no more? **Plain words:** from the package alone, state in at most five sentences what is being decided and what stays unknown. Is an evidence or instrument gap put to the decider as a question? Does the package state the run depth, the patterns relied on and the role register as the parts hold them? Does the value case hold any figure that is not an atom or a decision? Does each proposed priority follow the rule of Annex H.7?	A gap put as a question. Restatements that differ in substance between families: the text is unclear. The run depth, a reliance on a pattern or a role finding left out or misstated. A figure in the value case that is not an atom or a decision. A proposed priority that the rule does not give.
Point	**Stage**	**Packet**	**The engines are asked**	**MATERIAL means**
RP9 Release read	S8 Decide and release (before filing)	The frozen manifest; the decision references; the release parts; the release lint report	Does the release match every decision? Is every correction applied at its step? Is every known defect in the headline? Does any part say anything about how the	A mismatch with a decision. A correction not applied. A known defect missing from the headline. Any word about engines, agents, automation or tools that is not

Point	Stage	Packet	The engines are asked	MATERIAL means
			work was produced or checked, beyond the fixed forms of the quality checks? Is every lint exemption the organisation's own subject matter? Is every role-type the parts name in the role register?	the organisation's own subject matter. A role-type named in a part and missing from the role register. A MATERIAL finding stops filing; the release is re-frozen after repair, and was never filed.
RPa Needs pass	Asking service, before each pass is sent	The pass's rows; the do-not-ask list; the references searched; the Missing Report; the needs log	Does a held source already answer a row? Is each row one decision? Is a row leading, or does it name a person? Is each row's delegability right? Does the Missing Report equal the needs log?	A row a held source answers. A row that names a person. A bundled row. A non-delegable row marked delegable. A Missing Report that differs from the needs log. The pass is held; the offending rows are withdrawn or rewritten by their raising Cell, and RPa runs again.

Process (SOP)

Field	Content
Trigger	A step locks an output that has a review point, or a needs pass is ready.
Steps	see the numbered steps below the table
States	packet → drawn → blind pass → debate → settled → ruled → clear; or repair (cycle 1 or 2); or held → awaiting choice → chosen. Flags: waiting: ruling; held: no reviewers; held: review budget.
Failure and recovery	**The material finding notice** offers these choices, each with its cost. **SUPPLY:** the named missing fact-types are sought as needs requests, and the affected step re-runs when they are answered. **WITHDRAW:** the affected element or child is taken out of this release and entered in the deferred-boundary register. **REDESIGN:** the affected step re-runs from the smallest affected step, with a closure test sealed first. **ACCEPT AS KNOWN DEFECT:** the Requesting Authority accepts the finding as a declared known defect, in a non-delegable row; the element is released with the finding stated in the release's headline, in the Decision Package and in the walkthrough report. **CLOSE:** the run ends with a Held Report, filed into Source Lifecycle as a source of origin kind design report. The notice carries one recommended choice; none is adopted until the Requesting Authority decides. No element is released with a MATERIAL finding that is neither corrected, withdrawn, nor accepted as a known defect. Silence decides nothing (Annex C.1).
Records	Review records, ruling sheets, notices, choices and repair records, in the run record.
Cadence	per review point
Measures	Findings by class and review point; per review: level, upheld, withdrawn, void, contested; changes of position; dissent rate; rulings, with dismissals and downgrades; repairs; holds; calls and cost. No target is set.

1. **Chief of Staff (runs the packet code step)**: builds the packet of the review point; nothing of the dialogue, except at RP1b; on a re-review, no earlier finding → packet (evidence: hashes).
2. **Chief of Staff (runs the draw code step)**: draws seats and engines under rules 2–4, 17 and 18 → seats, engines and level (evidence: the gateway log; the engine register version).
3. **Blind Reviewers**: each reviews alone → blind cards (evidence: the gateway log shows every blind call ended before round 1).
4. **Chief of Staff (runs the check code step)**: quote and coverage checks; targeted pass → checked cards (evidence: the check log).
5. **Chief of Staff (runs the collation code step)**: merges findings, removes authorship and counts, fixes the form, shuffles → round-1 set (evidence: hashes).
6. **Blind Reviewers**: cross-examination → positions.
7. **Chief of Staff (runs the collation code step)**: builds the round-2 set → round-2 set.
8. **Blind Reviewers**: rebuttal → final positions.
9. **Chief of Staff (runs the settlement code step)**: settles every finding, computes the verdict, writes the review record → review record and verdict (evidence: a second run gives the same verdict; the label externally-audited, at its level, is written from this record only).
10. **Chief of Staff**: confirms upheld MATERIAL findings; sends contested ones to the Design Lead on ruling sheets → finding register.
11. **TOM Design Lead**: rules → ruling (evidence: signed with a live sign-in).
12. **Bounded repair** (rule 19) → repair record.
13. **Chief of Staff**: on a held step, sends one material finding notice as a decision needs request through Source Lifecycle → notice (evidence: needs log).

Role card

Field	Content
Name	Blind Reviewer A, B, C, D, E
Position and seat	Blind Reviewer; accountable seat the TOM Design Lead ([ORG]HOAC-ACS)
Performed by	agents ([ORG]CELL-TBRA to [ORG]CELL-TBRE), each with its own bot, credential and seat test. For each review the draw binds a drawn seat to one eligible engine of the engine register (§8). Any seat may sit in place of another.
Purpose	say whether the output claims more than its inputs support, and whether it keeps its rules
Authority limits	alone: write its own card and its own positions; never: see the work's dialogue (except at RP1b), or another engine's card before round 1; use a tool; review its own work; propose a design or a repair; rule on a finding
Write location	its own cards and positions
Reports to	the Chief of Staff
Escalation	to the Chief of Staff in one line; a stop goes to the stop door
Tools and access	none; no file, search or web tool; no credential in any instruction
Deputy and cover	none; any other seat covers an unavailable one
Fixed test and score	TEST-TOM-TBRA v5 to TEST-TOM-TBRE v5: cases 22, 53, 55 and 72, and the reviewer battery (§9)
Card version and status	v0.4.0 · draft

Fixed test

TEST-TOM-TR v5: TEST-TOM v5, cases 22, 23, 41, 53–64, 72, 88, 90, 91, 93, 94 and 97, and the reviewer battery · result NOT_RUN

5.9 Decision Package and Release Definition

Field	Content
Name	Decision Package and Release
What it is	The preparation of the decision for the Requesting Authority, in the organisation's own language. After the decision, the release of the model as one frozen, content-addressed **TOM Release**, filed back into Source Lifecycle as a source.
Purpose	Let the organisation decide its operating model element by element, without reading any technical file, and keep the decided model where every later stage will look for it.
Scope	Does: generate the quality checks; compile the value case and propose each element's priority, by code; write the Decision Package; send the decision rows; record the decision references; freeze and release; run the release lint; file the release; send the release notice to the next stage; carry earlier slices forward. Does not: decide; count silence or a general approval as a decision.
Inputs	The decided Design Brief and structure; the complete child designs, the hand-offs and the walkthrough, each with a settled review, or carrying only MATERIAL findings accepted as known defects; the quality checks (Annex E.11), generated by code.
Outputs	The **Decision Package** (docs/OUTPUT.md §10); its decision rows, as needs requests; the **TOM Release** (contracts/tom-release.schema.json); the release lint report (run record); the **filing receipt** (contracts/tom-filing-receipt.schema.json); the **release notice** to [ORG]AIOC-PPD (contracts/release-notice.schema.json).
Done when	Gates T8, then T9 (§5.0).

Field	Content
Limits	The decision rules of Annex C.4 apply. **What is put to decision.** An element is one of: a child's capability map as a whole (with each capability's delivery model and maturity target); a value stream (with whether it is critical); a decision domain; an entry of the hand-offs register; a counterparty class (with its journeys); an entry of the role register; the value case (Annex H.6). Each element row carries its proposed priority, and the decision records the priority decided (Annex H.7). The strategy frame is decided at S1 (Annex H.1). Before any row is sent, the Decision Package states the number of rows and passes. Where they exceed the configured limit (§15), rows are grouped one level up (per child, then per part), and the grouping is stated; one decision per row still holds. Evidence gaps and instrument gaps are never put to the Requesting Authority as questions. **Purpose and values.** Rows on the organisation's purpose and values are the listed type set-purpose-or-values (HF-04). The Cell carrying [ORG]HOAC-GOV signs them at the tool gateway's signature surface (Standard §6.1, item 6; Protocol §4.10, items 7-10), never through the Chief of Staff of [ORG]AIOC-SL or any Panel. Source Lifecycle records the decision with its signature_ref. This AIOC treats the row as decided only when that signature record is valid for the exact row payload. **Corrections.** A correction that states a fact about the organisation is admitted as organisation-stated by its decision reference, and checked like any other evidence. A correction that changes only a design choice is applied as a decision. A correction that needs a fact no source holds becomes a needs request, never an edit. **The release says nothing about how it was made**, beyond the quality checks. The release lint checks this by code (Annex D item 19; Annex E.15). **Slices:** see Slices and carry-forward, below. The TOM Design Lead attests that the method was followed; the attestation is not a design decision.
Sources	49GAION Standard v1.5.0 §4.4 items 6-9 (signature list, signer); §6.1 item 6; Operating Protocol v1.5.0 §4.9, §4.10 items 7-10, §13 items 5 and 6. Design basis: docs/DESIGN-BASIS.md §8; rule texts: docs/CARRIED-RULES.md §5, §12
Status	draft

Process (SOP)

Field	Content
Trigger	Every reviewed step of the slice is clear, or withdrawn by decision, or carries only MATERIAL findings the Requesting Authority accepted as known defects (§5.8).
Steps	see the numbered steps below the table
States	packaging → checked → awaiting decision → decided → released → filed · or returned for redesign · or rejected · or stopped: non-response (§5.1, failure).
Failure and recovery	A change after release is a new release that supersedes the old one, never a patch. Until the decision, no later stage starts on this model. A release lint hit, or a MATERIAL finding at RP9, stops filing: the release is repaired and re-frozen, and the frozen release that failed was never filed. Two failures on the same step stop the step and go to the TOM Design Lead.
Records	The package, the quality checks, the RP8 and RP9 records, the attestation, the decision references, the release manifest, the lint report and the filing receipt.
Cadence	per release
Measures	Rows decided per element; corrections; passes used; time from package to decision (measured, no target).

- **1. Chief of Staff**: writes the Decision Package (docs/OUTPUT.md §10). It contains: an executive summary; for each element, its purpose, what it owns and does not own, why it exists and its main evidence; value streams in plain language; the decision model; the boundary as inside, outside and deferred, visibly separated; the rationale; the open items in five blocks (A evidence gaps, B instrument gaps, C deferred cross-child scope, D remaining design uncertainty, E items needing a decision now, and only E asks anything); the missing-evidence signal in plain words; risks and limitations; the walkthrough summary; findings of the quality checks only where they help the decision, never wording hygiene; the elements carried from an earlier release, listed apart as decided in release <ID>; the sources superseded since the pin, in

plain words; the run depth, in plain words; the general patterns relied on, and each element resting on one alone; the role register in plain words, with each role finding and how it was settled; the strategy frame, with each element not given; each capability's delivery model and maturity, and each critical stream with its reasons; each outside party's journey; the value case; how the work runs today, or that no record of it was given; each element's proposed priority, with the rule that gave it; the number of rows and passes → package (evidence: hash).

- **1b. Chief of Staff (runs the value-case and priority code step)**: compiles the value case from the decided parts and the decision references (Annex H.6), and proposes each element's priority by the fixed rule of Annex H.7 → value case; proposed priorities (evidence: hashes; mechanically-verified).
- **1a. Chief of Staff (runs the quality-checks code step)**: generates the quality checks (Annex E.11) from the settled review records and rulings → quality checks (evidence: hash).
- **2. Review Pod**: review point RP8 (§5.8). Every statement traces to a reviewed part and claims no more; the plain-words restatement → RP8 record (evidence: externally-audited). Once RP8 has settled, the quality-checks code step adds the section of this stage from the settled RP8 record.
- **3. TOM Design Lead**: attests that gates T0-T7 passed, each with its evidence → attestation (evidence: run record, live sign-in).
- **4. Chief of Staff**: sends the decision sheet as decision needs requests: one row per element (ACCEPT, CORRECT with a comment, REJECT), each showing its proposed priority (must, should, could), which ACCEPT keeps and CORRECT may change (Annex H.7); one row for the value case; then one final-status row (ACCEPT; ACCEPT WITH CORRECTIONS, before or after the next stage; RETURN FOR REDESIGN; REJECT). Carried elements are not asked again. Purpose-and-values rows are marked with their listed type → decision rows (evidence: needs log).
- **5. Requesting Authority** (in Source Lifecycle, outside this AIOC): decides each row. The Cell carrying [ORG]HOAC-GOV signs the purpose-and-values rows at the gateway. Source Lifecycle records each decision → decision references (evidence: Source Lifecycle's decision records).

- **6. Chief of Staff**: applies corrections at the smallest affected step and re-runs only what they re-open, with the failing situations as regression. On ACCEPT or ACCEPT WITH CORRECTIONS, it freezes the release: the manifest names every part and every run record by hash, every pack version with its snapshots, the decision references and every version → TOM Release (evidence: the release_id, computed over the manifest without any filing data).
- **6a. Chief of Staff (runs the release-lint code step)**: lints every part (Annex D item 19). A hit fails T9 → lint report (evidence: the report's hash in the manifest).
- **6b. Review Pod**: review point RP9 (§5.8) → RP9 record. A MATERIAL finding stops filing; the release is re-frozen after repair.
- **7. Chief of Staff**: files the TOM Release into [ORG]AIOC-SL as a source of origin kind operating model. It files through the write scope of the connection, on behalf of the TOM Design Lead's Cell, whose position has accepted the contributor agreement. The filing carries the final-status decision reference as its approval; Source Lifecycle sets the release's status from that decision, and nothing more is asked of the Requesting Authority → filing receipt (evidence: the receipt validates; the filed source reads back byte-identical by hash).
- **8. Chief of Staff**: sends the release notice to the Requesting Authority's Cell through the HU output port. The run Work Item is done only when the Requesting Authority's acceptance test passes on the filed reference (Protocol §4.5) → notice (evidence: Office record).
- **8a. Chief of Staff**: sends the release notice to the next stage, [ORG]AIOC-PPD, through the gateway on the connection of §6 (Protocol §19, item 5). It validates against contracts/release-notice.schema.json (Annex E.20): the release ID, version and manifest hash; the filed reference; the final-status decision reference; the release it supersedes; and the returns it answers (§5.10). It carries no release text: the next stage reads the release from Source Lifecycle by that reference. It is sent once per release, only after the filing receipt reads back. Where the connection is not signed, the notice waits, shown on the Panel; nothing of the run waits for it → release

notice (evidence: the Work Item, and the notice's hash in the run record). **New** in v0.5.0.

Slices and carry-forward. A release covers one slice. The next slice is designed in a continuation run, by these rules:

1. **Entry.** A continuation run's Design Request (DRQ-01 field 8) names the release it continues and any children to change. P0 lists that release under carried, with the pack IDs named in its manifest.
2. **Carried as decided.** Each carried child enters decided, by hash. Its child brief, package and design ledger are admitted at T3, T4 and T5 without re-design, with their decision references, unless rule 3 applies.
3. **Re-entry.** A carried child re-enters at its smallest affected stage when the new Design Request names it for change. It re-enters at the first stage whose input changed (S4 at the earliest) when a source under its load-bearing elements changed: an atom superseded, revoked or reclassified, or no longer resolving by hash at its snapshot. Otherwise it stays decided.
4. **Party register.** The carried children's marker maps are reconciled with the new party register. A marker that now names a new sibling child is re-paired in Hand-offs (the carry-forward record, Annex E.16).
5. **Seams.** Hand-offs pairs every crossing of a carried child with the new children. A seam that does not pair is routed to the new child's Structure, Package or evidence, never to the carried child, unless the carried child is named for change (rule 3). A seam that cannot be closed on the new side becomes a decision row.
6. **Walk.** The walk covers the whole set of children, carried and new. Every earlier failing situation, and every earlier regression set, re-runs as regression. A fresh situation set never erases a recorded failure.
7. **Decision.** Carried elements are not asked again. The Decision Package lists them apart, as "decided in release <ID>", and asks for the new and changed elements and one final status for the whole release.
8. **Release.** The new release names every carried part by its original hash (parts[].carried_from), lists the carried release under carries and its pack versions under input_packs (with from_release), contains every slice designed so far, and supersedes the earlier release (supersedes).

9. **Trace.** T9 accepts an atom of a carried release's pack version only if it still resolves by hash at its snapshot (SL-A8, as amended).
10. **A return from a later stage** arrives directly and ends by §5.10 (CARRIED-RULES.md §4.4). Where the Requesting Authority answers it by a release, it opens a continuation run whose revised Design Request cites the return's filed copy in Source Lifecycle, and the new release's notice names the returns it answers.

Role card

Field	Content
Name	Chief of Staff (decision and release)
Position and seat	Chief of Staff; accountable seat the TOM Design Lead
Performed by	an agent ([ORG]CELL-COSTOM); the attestation is by the TOM Design Lead (§3)
Purpose	put the decision to the Requesting Authority honestly, and release and file exactly what was decided
Authority limits	as in §4.2; never: decide; send a row before RP8 has settled clear; file a release whose final status is not decided, whose release lint failed, or whose RP9 is not settled
Write location	the run record; the release; the filing
Reports to	the TOM Design Lead
Escalation	as in §4.2
Tools and access	as in §4.2, plus the write scope of the connection, for origin kinds operating model and design report only
Deputy and cover	none; the run halts
Fixed test and score	TEST-TOM-COSTOM v5 (cases 23, 24, 27, 36, 51, 52, 64, 67-69, 80, 82, 85, 98, 117, 118, 122)
Card version and status	v0.6.0 · draft

Fixed test

TEST-TOM-COSTOM v5 (cases 23, 24, 27, 36, 51, 52, 64, 67-69, 80, 82, 85, 98, 100, 117, 118 and 122 for this capability) · result NOT_RUN

5.10 Returns from later stages

New in v0.5.0 (decisions/DR-0005).

Definition

Field	Content
Name	Returns from later stages
What it is	The receiving end of the Library's design return contract (Protocol §19, items 5 and 6; 49GAION/DR-0016, 49GAION/DR-0018). A later stage, [ORG]AIOC-PPD or [ORG]AIOC-DI, sends a return on elements of a release this AIOC filed, directly. This capability admits the return, shows it, puts it to the Requesting Authority and carries it, round by round, to one of its ends.
Purpose	No defect a later stage finds is lost, and no return is left open: each ends, so that the stage after this one can deliver.
Scope	Does: admit and check each return; record it in the returns log; show it on the Panel with its age; generate its return brief; send its decision rows through Source Lifecycle; count its rounds and deadlines; escalate; record its end on the return and send it back to its sender; give each ended return to the TOM Design Lead to classify. Does not: change a filed release; decide or answer a return; argue a return away; judge the sender's work.
Inputs	A return (library/contracts/design-return.schema.json of 49GAION-protocol, at the version pinned in contracts/PINS.json), received through the gateway on the connection with its sender; its filed copy in Source Lifecycle, by reference; the release it names.
Outputs	The returns log (Annex E.21); the return brief and its decision rows (Annex E.22); the return at each next version, with its status (the return answer); the TOM Design Lead's classification of each ended return.
Done when	The return has ended: answered by release, answered by decision, set aside by decision with its carried_forward_on, or withdrawn by its sender; the ended return is sent back to its sender and its end is in the returns log; the TOM Design Lead has classified it.

Field	Content
Limits	1. **Admission.** A return is admitted only if it validates against the pinned contract; the release it names is a release this AIOC filed, with the same manifest hash; its filed_ref reads back from Source Lifecycle with the same hash; and it came on the connection of its sender. Otherwise it is refused back to its sender with the reason (Protocol §4.4), never edited. 2. **Never a change to the release.** A filed release is never edited. A return is answered only by a new decided release, made by a continuation run (§5.9, Slices and carry-forward, rule 10), or by a decision of the Requesting Authority. 3. **accepted asks nothing.** An accepted return is recorded and ends at once, answered by release, with answered_by the release it names. 4. **A brief made by code; one decision per row.** The return brief is generated by code from the return and the release elements it names, in plain words; nothing is added by hand. It offers the Requesting Authority three rows, each decided alone: open a continuation run, naming the children to change; answer by decision (a correction that states a fact about the organisation is admitted only as §5.9, Limits, Corrections, says); or set the return aside and carry its elements forward as a known defect or a decided assumption. Silence, or a general approval, decides nothing. No row names a person. 5. **Rounds.** A return opens in round 1, with the deadline return_round_deadline (§15). Each answer that does not end it opens the next round with its own deadline: an answer by release opens the next round when the sender returns the same elements again, under the same return_id and the next round. 6. **Escalation.** Past a round's deadline, or at return_rounds_max, the return goes to the Requesting Authority with two rows only: answer by decision, or set aside on a stated basis. It is raised to the Cell carrying [ORG]HOAC-TM and shown in red on both Panels until it is decided. 7. **Only the sender's touched elements wait.** The elements the return names in held_here wait at its sender; this AIOC holds no run because of a return. 8. **Classified, never self-graded.** Each ended return is classified by the TOM Design Lead, never by the Chief of Staff: a defect of this AIOC (information it should reasonably have obtained, docs/CARRIED-RULES.md §4.4, item 3), or normal discovery. A defect becomes a lesson in §12 and a proposed test case (H-18).

Field	Content
Sources	49GAION/DR-0016, items 4 and 5; 49GAION/DR-0018; Protocol §19, items 5 to 7, and §20, item 3; library/contracts/design-return.schema.json of 49GAION-protocol; docs/CARRIED-RULES.md §4.4
Status	draft

Process (SOP)

Field	Content
Trigger	A return arrives on the connection of [ORG]AIOC-PPD or [ORG]AIOC-DI; a round's deadline passes; a decision on a return row is recorded; a release that answers a return is filed.
Steps	see the numbered steps below the table
States	the contract's: open (round n, with its deadline) → answered by release · answered by decision · set aside by decision · withdrawn; and the flag escalated, with its age
Failure and recovery	A return that does not validate is refused with its reason; its sender may send it again. A revoked connection holds the sending of answers, never the return itself. Two failures on the same step go to the TOM Design Lead.
Records	The returns log; each return's versions; the return briefs; the decision references; the classification records.
Cadence	per return
Measures	Returns by sender and kind; rounds per return; age at end; ends by kind; returns classified as defects of this AIOC (measured, no target).

- **1. Chief of Staff**: admits the return by limit 1, or refuses it with the reason → admitted return or refusal (evidence: the returns log, with the return's hash).

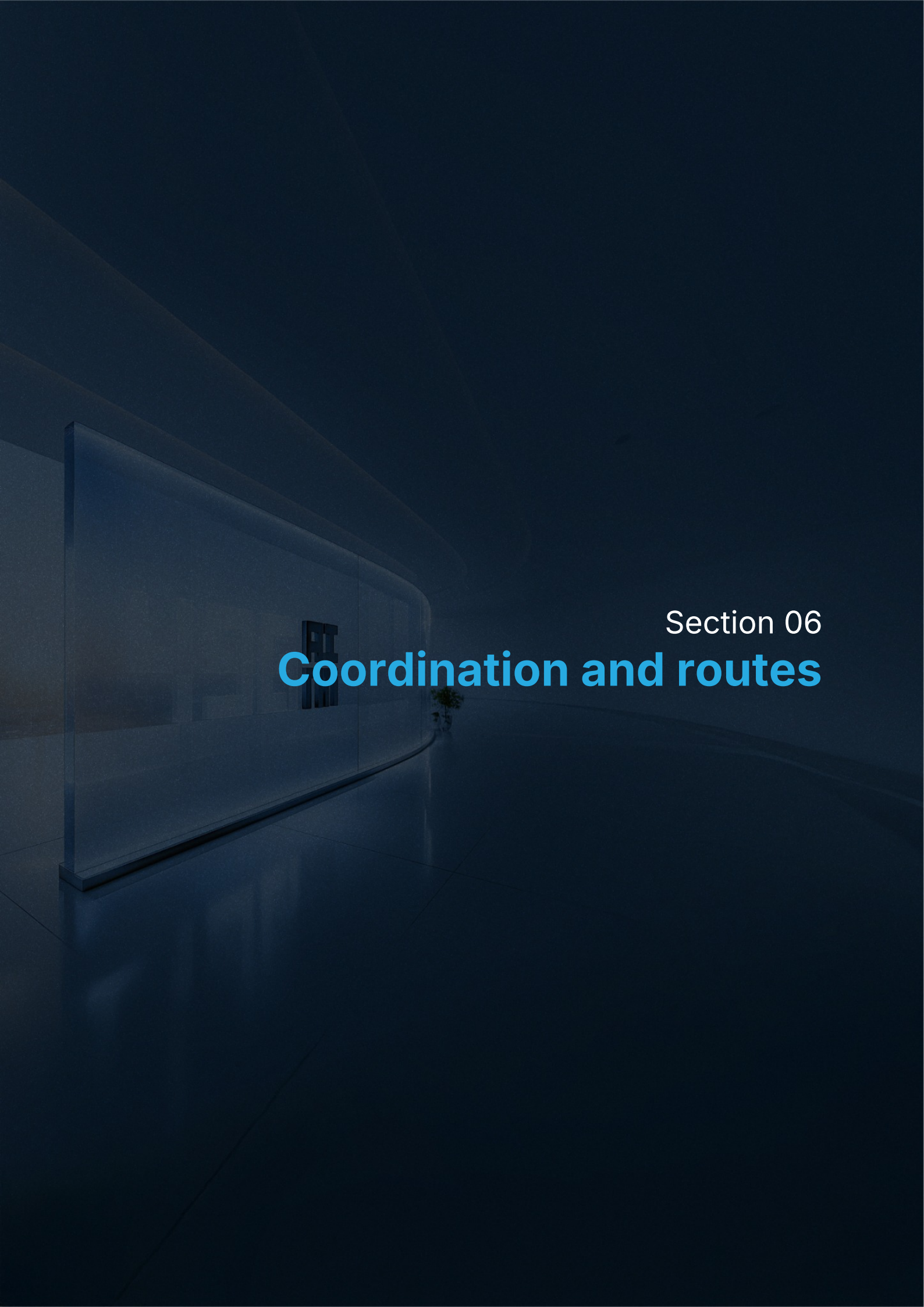
- **2. Chief of Staff**: records it in the returns log and shows it on the Panel with its age; ends an accepted return at once (limit 3) → recorded return (evidence: the returns log).
- **3. Chief of Staff (runs the return-brief code step)**: generates the return brief and its rows from the return and the release elements it names (limit 4) → return brief (evidence: hash).
- **4. Chief of Staff**: sends the rows as decision needs requests through [ORG]AIOC-SL (§5.2), one decision per row, with the brief as their explainer → decision rows (evidence: the needs log).
- **5. Requesting Authority** (in Source Lifecycle, outside this AIOC): decides one row. Source Lifecycle records the decision → decision reference (evidence: Source Lifecycle's decision record).
- **6. Chief of Staff**: records the decision on the return as its next version. answered by decision, or set aside by decision with its carried_forward_on, ends it. A decision to open a continuation run keeps it open until the new release is filed; that release's notice names the return, and the return becomes answered by release, with answered_by the new release ID → return answer (evidence: the return's next version, valid against the pinned contract).
- **7. Chief of Staff**: sends the return answer back to its sender, on the same connection → sent answer (evidence: the Work Item).
- **8. Chief of Staff**: on the sender's next round of the same return, repeats steps 2 to 7; at a deadline passed, or at return_rounds_max, escalates by limit 6 → escalated return (evidence: the returns log).
- **9. TOM Design Lead**: classifies each ended return by limit 8 → classification record (evidence: run record, live sign-in).

Role card

Field	Content
Name	Chief of Staff (returns)
Position and seat	Chief of Staff; accountable seat the TOM Design Lead
Performed by	an agent ([ORG]CELL-COSTOM); the classification is by the TOM Design Lead (§3)
Purpose	carry every return to an end, honestly, without changing a release
Authority limits	as in §4.2; alone: admit or refuse a return by limit 1, record it, run the return-brief code step, send its rows and its answer; never: decide, answer or classify a return, edit a release, write a brief by hand, let a return wait without its round and deadline
Write location	the returns log; the return's versions
Reports to	the TOM Design Lead
Escalation	as in §4.2; a return past its deadline goes to the Requesting Authority and the Cell carrying [ORG]HOAC-TM (limit 6)
Tools and access	as in §4.2: the connections with [ORG]AIOC-PPD and [ORG]AIOC-DI and with Source Lifecycle
Deputy and cover	none; returns wait, each shown on the Panel with its age
Fixed test and score	TEST-TOM-COSTOM v5 (cases 101–108)
Card version and status	v0.5.0 · draft

Fixed test

TEST-TOM-COSTOM v5 (cases 101–108 for this capability) and TEST-TOM-TDL v5 (TDL-8) · result NOT_RUN



Section 06

Coordination and routes

- **[ORG]AIOC-SL.** It is the only source this AIOC asks and files into, and the only route to any person in the organisation.
  - **Routes.** Sources, answers and status messages arrive by reference through the Input socket. Needs requests, Missing Reports, status messages and filings leave as Work Items through the output port.
  - **The organisation-design connection.** It is signed by the [ORG]HOAC-ACS seat that [ORG]AIOC-SL is accountable to, as connect-external-port (Protocol §13, item 5); a wider scope is grant-or-widen-access. It has three parts:
    - unrestricted read across every collection and confidentiality level, as Source Lifecycle requires for an organisation-design connection;
    - least-privilege write, only for sources of origin kind operating model or design report;
    - needs requests.
  - **Expiry and revocation.** Scope expiry is checked on every read. A revocation cuts the connection at the tool gateway at once: every later read is refused, and the run is held as held: connection revoked (Annex C.1). A revocation is not a stop, and no restart is needed.
- **Other source types** are read only, each through its own adapter (§4.3), each connection signed by the seat over that source as connect-external-port.
- **Work Items.**
  - **The run.** A run is one Work Item, opened by the Requesting Authority's Cell through this AIOC's HU input port. Its fields:
    - requested output: a TOM Release under contracts/tom-release.schema.json;
    - acceptance test: the Requesting Authority's (gate T8 decided; the manifest valid; the filed reference reads back);
    - sources: the Design Request reference.
  - **Needs requests.** Each is a sub-item of the run, with chain depth 1. It goes to [ORG]AIOC-SL, and its requested output is a needs answer (Protocol §4.6,

item 2). Source Lifecycle answers the sub-item; it never opens a new item back to this AIOC, so the chain never loops.

- **[ORG]AIOC-PPD, the next stage, linked directly** (Protocol §19, items 5 and 7). **New** in v0.5.0.
  - **Routes.** On every filing of a decided release, a release notice leaves through the output port (§5.9 step 8a). PPD reads the release itself from Source Lifecycle, by the reference in the notice. Its returns arrive at the Chief of Staff's port, and end by §5.10.
  - **The connection.** One connection with two scopes: release notices to PPD; returns from PPD and their answers. It is signed by the [ORG]HOAC-ACS position over this AIOC, the TOM Design Lead's, as connect-external-port (H-35). A wider scope is grant-or-widen-access. A revocation cuts it at the gateway at once; notices and answers wait, shown on the Panel. This AIOC stands alone without it (Protocol §19, item 9).
  - **Loops.** A return is a Work Item from PPD. This AIOC answers it on that Work Item and never opens a new item back, so the chain never loops (Protocol §4.6, item 2).
- **[ORG]AIOC-DI, the last stage** (Protocol §20, item 3). **New** in v0.5.0. Its returns on elements of this AIOC's releases arrive and end in the same way, on a connection with one scope (returns from DI and their answers), signed as above (H-35). Nothing else is sent to it; it reads this AIOC's releases from Source Lifecycle, by reference.
- **[ORG]AIOC-OCO.** It installs, fixes and upgrades this AIOC through the front door. Every proposal, correction or critique of this AIOC's AIs is a design-proposal Work Item to OCO.
- **Routing rows this AIOC contributes.** needs request → [ORG]AIOC-SL; missing report → [ORG]AIOC-SL; needs status → [ORG]AIOC-SL; TOM Release filing → [ORG]AIOC-SL; release notice → [ORG]AIOC-PPD; return answer → the sender of the return.



Section 07

Records and where things live

Record	Home	Source rule
This blueprint, its contracts, tests, decisions, design basis and carried rules	the writing repository <org>aioc-tom	one home for each thing
Adapters, their fixtures and tests	the writing repository (adapters/)	one home for each thing
The code steps of this blueprint, written new; the PRD; the private register of the earlier chain; the earlier chain's code under legacy/ as a case study, read only	the code project <org>aioc-tom	one home for each thing
The Input Pack versions, review records, ruling sheets, the party register, the lens record, the lint report, the run preview, the two drafts of S1 and S5 and their difference registers	<org>aioc-tom-records; -records-restricted as below	append-only; referenced from the release manifest by hash
An organisation's run records: read plans, round logs, ledgers, gate reports, review records, traces	<org>aioc-tom-records; -records-restricted for any run record that touches a Strictly confidential source or personal data	append-only

Record	Home	Source rule
The organisation's sources, the answers to needs requests, the decisions, the filed TOM Release	[ORG]AIOC-SL	Source Lifecycle is the only home of a source
Work Items	the shared queue	Protocol §2

- **References, not copies.** Atom text is held only in the run workspace, resolved through its adapter by <adapter ID>:<source ID>@<version>#<anchor> and hash. Records store atom IDs and hashes, never atom text.
- **Revocation or reclassification.** On a notice from Source Lifecycle (docs/INTAKE-WITH-SOURCE-LIFECYCLE.md §9), the run is held and its workspace copies are sealed. A supersession after the pin flags the affected atoms in the next pack version (§4.3).
- **Engine ceilings.** No atom is sent to an engine whose confidentiality ceiling is below the atom's level (§5.8 rule 17; Annex E.12).
- **Confidentiality of the release.** The release quotes a source only where that source's confidentiality level allows it under the grant. It takes the highest confidentiality level of anything it quotes, and Source Lifecycle classifies it so.



Section 08

Tools and the shared AI gateway

Tool	Existing shared tool, or new	If new: asked of the builder as
The Source Lifecycle connection (output socket, needs requests, filing)	existing in [ORG]AIOC-SL; the needs-request path, the added reference fields (SL-A8 as amended) and filing by a connected part are proposed additions (docs/INTAKE-WITH-SOURCE-LIFECYCLE.md §9)	a Work Item to the Source Lifecycle technical lead
The code steps: the socket (identify, normalise, atomise, label, pack builder, pack report); prepackaging, package assembly, gates G1-G5; design checks, renderer; pairing, seam check; walk-input compile, blind-brief extract, compile, seal, walk; review (packet, draw, quote and coverage check, collation, settlement, packet split); quality-checks render; release lint; carry-forward reconcile; draft compare; role-register compile; pattern-reference check; release notice; return admission, return brief and the round clock (§5.10)	new in this AIOC's code project, written against this blueprint and docs/CARRIED-RULES.md, one build item per step, each with its tests. The earlier chain's code is kept unchanged under legacy/ as a case study: read, never wired in, never copied as a starting point.	build items in the code project

Tool	Existing shared tool, or new	If new: asked of the builder as
The schemas between Pods (design brief, structure, child package, packaging ledger, design ledger, finding register, decision package, difference register, role register, and the run records of Annex E), each validated by producer and consumer	field lists closed in Annex E and docs/CARRIED-RULES.md; JSON Schemas made from them may add nothing; build items before the first run	build items in the code project
The standard situation library	new; a build item before the first real walk	a build item in the code project
Engine calls	through the tool gateway, which binds each call to an engine of the engine register, meters it and logs the engine served	—
The engine register	at the tool gateway (Annex E.12)	a change-gateway-policy (H-05)
The reference-model catalogue	a Library item of 49GAION, proposed (upstream/REFERENCE-MODEL-CATALOGUE.md); read by version and hash (contracts/PINS.json)	a pull request to 49GAION (H-09), then a pin (H-34)

- **Metering.** Every engine call goes through the tool gateway of the [ORG]AION, metered for this AIOC (Standard §4.4, item 11).
- **Engines and seats.** Each AI Cell's filler is its bot, with its own identity card and credential. The gateway binds each call to an engine of the engine register and logs the engine served. For each review, the draw binds each drawn reviewer seat to one eligible engine (§5.8). Binding an engine is not a change of filler. An engine may serve a Cell only after passing that Cell's fixed test at its current version (Standard §3.2 item 2; §4.4 item 4). That

binding an engine is not a change of filler is this AIOC's reading of the Standard, proposed to 49GAION as a reading note (upstream/PROPOSALS.md, item 6).

- **New. Null default.** Every AI Cell runs with no tools, no outside access, no memory and a fresh context for each dispatch. Any grant above that needs a rationale, a scope, a permission level and its effect on qualification.
- **New. Instruction packs.** They are generated from this blueprint and checked by diff; hand-paraphrase is forbidden.
- **New. Tests stay hidden from the runners.** No Cell under test sees TEST-TOM, its fixtures, the review protocol, §9 or the reviewer battery.

Tool role	Connector	Access managed in	Part card version
source of record	[ORG]AIOC-SL output socket	Panel (the unit card schema fixes this value): the Panel shows the connection and requests it; the grant itself is signed at the tool gateway as connect-external-port by the [ORG]HOAC-ACS seat over [ORG]AIOC-SL (upstream/PROPOSALS.md, item 4)	49GAION Source Lifecycle, v1
next design stage	[ORG]AIOC-PPD input port: release notices out, returns in	Panel: the Panel shows the connection and requests it; the grant is signed at the tool gateway as connect-external-port by the [ORG]HOAC-ACS position over this AIOC (H-35)	49GAION Process and Position Design, v2
returns from the last stage	[ORG]AIOC-DI output port: returns in	as above	49GAION Delivery and Implementation, v1



Section 09

Fixed tests: index, triggers and scores

Each part has its own fixed test, made of the cases of TEST-TOM v5 (tests/TEST-TOM.md) that exercise it. The owner of every part's test is the TOM Design Lead ([ORG]CELL-TDL). The owner of the AIOC's whole test, and of the Design Lead's own seat test, is the Cell carrying [ORG]HOAC-TM.

Part	Test ID and version	Cases of TEST-TOM v5	Evidence level	Last run	Result
[ORG]AIOC-TOM (whole), with path qualification	TEST-TOM v5	all (1–123), path qualification, the walkthrough battery, the reviewer battery	MEASURED	—	NOT_RUN
[ORG]CELL-TDL TOM Design Lead	TEST-TOM-TDL v5	TDL-1 to TDL-8	MEASURED	—	NOT_RUN

Part	Test ID and version	Cases of TEST-TOM v5	Evidence level	Last run	Result
[ORG]CELL-COSTOM Chief of Staff (with Office, Input socket, Needs Requests, Decision Package and Release)	TEST-TOM-COSTOM v5	1–13, 23–36, 40, 43, 44, 46, 49–52, 61, 64, 67–69, 75, 76, 78, 80–82, 85, 87, 89, 90, 92, 94–96, 98–110, 117–122	MEASURED	—	NOT_RUN
Input socket adapter ADP-SL	TEST-TOM-ADP-SL v2	its fixtures (adapter file §9); cases 47, 48, 119	MEASURED	—	NOT_RUN
Panel	default Panel test	—	MEASURED	—	NOT_RUN
[ORG]POD-TRR Request Reconstruction	TEST-TOM-TRR v5	4, 5, 8, 14, 15, 28, 73–76, 83, 91–93, 109–111, 120, 121, 123	MEASURED	—	NOT_RUN

Part	Test ID and version	Cases of TEST-TOM v5	Evidence level	Last run	Result
[ORG]CELL-TSP Source Proxy	TEST-TOM-TSP v5	4, 5, 14, 92	MEASURED	—	NOT_RUN
[ORG]CELL-TRA Request Analyst	TEST-TOM-TRA v5	8, 14, 15, 28, 73–75, 83, 91, 92, 109–111, 123	MEASURED	—	NOT_RUN
[ORG]CELL-TSRA Second Request Analyst	TEST-TOM-TSRA v5	8, 14, 15, 73–75, 83, 91, 92, 109–111, 123	MEASURED	—	NOT_RUN
[ORG]CELL-TFR Frame Reconciler	TEST-TOM-TFR v5	93, 94	MEASURED	—	NOT_RUN
[ORG]POD-TS Structuring	TEST-TOM-TS v5	16, 42, 83	MEASURED	—	NOT_RUN

Part	Test ID and version	Cases of TEST-TOM v5	Evidence level	Last run	Result
[ORG]CELL-TWS Work Structurer	TEST-TOM-TWS v5	16, 42, 83	MEASURED	—	NOT_RUN
[ORG]POD-TEP Entity Packaging	TEST-TOM-TEP v5	17, 41, 45	MEASURED	—	NOT_RUN
[ORG]CELL-TP Packager	TEST-TOM-TP v5	17, 41, 45	MEASURED	—	NOT_RUN
[ORG]POD-TEM Entity Model Design	TEST-TOM-TEM v5	18, 19, 38, 39, 91–93, 96, 97, 112–115	MEASURED	—	NOT_RUN
[ORG]CELL-TD Designer	TEST-TOM-TD v5	18, 19, 38, 39, 91, 92, 97, 112–115	MEASURED	—	NOT_RUN

Part	Test ID and version	Cases of TEST-TOM v5	Evidence level	Last run	Result
[ORG]CELL-TSD Second Designer	TEST-TOM-TSD v5	18, 19, 38, 39, 91, 92, 97, 112–115	MEASURED	—	NOT_RUN
[ORG]CELL-TLR Ledger Reviser	TEST-TOM-TLR v5	18, 93	MEASURED	—	NOT_RUN
[ORG]POD-TH Hand-offs	TEST-TOM-TH v5	20, 37, 42, 65, 66, 70, 85–88, 113, 116	MEASURED	—	NOT_RUN
[ORG]CELL-THA Hand-off Analyst	TEST-TOM-THA v5	20, 37, 42, 65, 66, 70, 87, 116	MEASURED	—	NOT_RUN
[ORG]POD-TW Operational Walkthrough	TEST-TOM-TW v5	21, 71, 84, 91, the walkthrough battery	MEASURED	—	NOT_RUN

Part	Test ID and version	Cases of TEST-TOM v5	Evidence level	Last run	Result
[ORG]CELL-TSA Situation Author	TEST-TOM-TSA v5	21, 84, 91	MEASURED	—	NOT_RUN
[ORG]CELL-TSW Situation Walker	TEST-TOM-TSW v5	21, 71	MEASURED	—	NOT_RUN
[ORG]POD-TR Independent Review	TEST-TOM-TR v5	22, 23, 41, 53–64, 72, 88, 90, 91, 93, 94, 97, 111, the reviewer battery	MEASURED	—	NOT_RUN
[ORG]CELL-TBRA Blind Reviewer A	TEST-TOM-TBRA v5	22, 53, 55, 72, the reviewer battery	MEASURED	—	NOT_RUN
[ORG]CELL-TBRB Blind Reviewer B	TEST-TOM-TBRB v5	22, 53, 55, 72, the reviewer battery	MEASURED	—	NOT_RUN

Part	Test ID and version	Cases of TEST-TOM v5	Evidence level	Last run	Result
[ORG]CELL-TBRC Blind Reviewer C	TEST-TOM-TBRC v5	22, 53, 55, 72, the reviewer battery	MEASURED	—	NOT_RUN
[ORG]CELL-TBRD Blind Reviewer D	TEST-TOM-TBRD v5	22, 53, 55, 72, the reviewer battery	MEASURED	—	NOT_RUN
[ORG]CELL-TBRE Blind Reviewer E	TEST-TOM-TBRE v5	22, 53, 55, 72, the reviewer battery	MEASURED	—	NOT_RUN

- **Re-run triggers.** The template defaults, plus any change to a contract in contracts/, to the Source Lifecycle output socket, or to the situation library; and any change to an adapter file or its fixtures, to the engine register, or to the release lint deny-list; any change to the reference-model catalogue pin; any change to the pin of the Library's design return contract (contracts/PINS.json).
- **Where results are kept.** In this section; on the Panel's Health section; and, as evidence, in <org>aioc-tom-records.



Section 10

Build: what is generated from this file

Template defaults. In addition:

- **Validators.** The code project builds, from contracts/, the validators for every contract.
- **Cards.** tools/make_cards.py generates the unit cards and the bot identity cards in cards/ from §17 of this file, and checks them. It writes 26 unit cards and 18 bot identity cards. The blueprint governs; the generated cards are copies. It also checks every code against Naming Standard R15 and R16 and the withdrawn codes (§17).
- **Contract checks.** tools/check_contracts.py validates every contract and example; checks the content addresses of the release and the Input Pack; checks that the listed-type enums are a subset of the pinned signature list (contracts/PINS.json); checks each adapter's fixtures against its mapping block; and runs the release lint over docs/OUTPUT.md, the release enum strings and the lint fixtures; validates the reference-model catalogue seed in upstream/ and checks its pin (check 10); checks the pin of the Library's design return contract and validates this AIOC's return examples against it (check 11).
- **Release lint.** tools/release_lint.py is the release-lint code step of S8, and the CI check (Annex D item 19; Annex E.15).



Section 11

Exceptions

1. **The Chief of Staff never asks the organisation's people anything.** The default Chief of Staff carries requests to and from humans. This one talks only to its own HOAC seat, and only about the run. It reaches the organisation only through the Chief of Staff of [ORG]AIOC-SL. Reason: the direction recorded in decisions/DR-0001, that this AIOC works only from documents, because spoken words cannot be the basis of an operating model.
2. **The Input socket has one adapter in this version**, for Source Lifecycle (ADP-SL). Any other source type enters only through its own adapter file, read only (§4.3). Words typed or spoken to this AIOC are refused as input. Reason: as in item 1.
3. **The Back office does not converse with humans** (converses_with_humans: false on the card). Reason: as in item 1.
4. **The HU input ports of the Pods and inner Cells accept only run control and configuration from the TOM Design Lead**, never content about the organisation. Reason: as in item 1.



Section 12

Lessons and proposals to the mother

Lessons from the case study are history, not evidence (§0).

Date	Lesson	Test case it added	Proposed to the mother
2026-10-14	In the case study, a run's own checks missed material defects that seats outside the run caught. Prose checks became declared data checked by code.	18, 22	yes (this text)
2026-10-14	A walk's own classification of its findings can run toward leniency; the findings are now reviewed at RP7 under the Review Panel rule.	21, 22	yes
2026-10-14	Without the organisation's own approved statements as evidence, a design can stop only at the coarsest grain its documents allow. The organisation's statements, approved and recorded as sources, are now admitted with their own label.	31	yes
2026-10-14	No capability owned the cutting of sources into atoms; the work fell between three. It is now a code step of this AIOC.	17	yes

Date	Lesson	Test case it added	Proposed to the mother
2026-10-14	Review of v0.1.0: as first written, confirming an inference lowered its label; a run could not start for want of a defined Design Request; the frame went to decision before its independent review; the Chief of Staff ruled on findings about its own run. Each is fixed in v0.2.0.	3, 12, 15, 22	yes
2026-10-15	One pinned snapshot cannot hold the answers and decisions recorded after it; the Input Pack is versioned, and the trace reads every version	49, 51	yes
2026-10-15	Labels need atoms at admission; atomising and labelling moved into the Input socket	45	yes
2026-10-15	One or two reviewers of one family share blind spots; every stage is now checked by three or more engines from two or more families, who debate; code settles; a person rules on what stays contested	53–64	yes
2026-10-15	The release had described work it does not do; the release lint now keeps the release to the operating model	67	yes

Date	Lesson	Test case it added	Proposed to the mother
2026-10-15	Capabilities dropped because they were never developed, or because they needed conversation, can run through Source Lifecycle: the request kind, the lens, the understanding check, the evaluator's questions and Gate B are restored	72–76	yes
2026-10-15	A design read by one author keeps that author's blind spots, and the reviews of v0.3.0 proposed four features for later; the governing seat directed that they be added now: two authors at S1 and S5, a role register, a run depth, and reference models used only as methods	85–99	yes
2026-10-16	A later stage that could only file its returns in Source Lifecycle was not heard until someone opened a new run, and nothing made a return end. The next stage is now told of each release directly, its returns come back directly, and each ends in one of four ways	100–108	yes (49GAION/DR-0016, DR-0018)

Date	Lesson	Test case it added	Proposed to the mother
2026-10-16	A study of fifteen comparable products found that a decided model was still missing what the stages after it need to start: the strategy it serves, how each capability is delivered and how mature it is, which streams are critical, each outside party's journey, the value case, a priority per element, and how the work runs today. Each is added in the stage where it belongs, from sources and decisions only, never written by this AIOC and never about a person	109–123	yes (49GAION/DR-0023)



Section 13

For the [ORG]HOAC-ACS seat, and every decision only a human seat can take

Every decision that only a human seat can take is listed here once, with its seat and its default. Silence decides nothing; a general approval decides no row.

ID	Decision	Seat	Default
H-01	Name the Requesting Authority's Cell: the GOV Cell itself, or a human Cell to which it delegates the design decisions (delegate-authority)	GOV	none: T0 fails
H-02	Withdrawn (49GAION/DR-0049): a compensating control for one person behind the TOM Design Lead and the Requesting Authority's positions is the organisation's own; nothing in this AIOC records or reads it	—	—
H-03	Sign the purpose-and-values rows (set-purpose-or-values, HF-04) at the gateway's signature surface	GOV	undecided until signed
H-04	Add accept-operating-model to the organisation's own signature list (change-org-standard-or-signature-list; Protocol §4.9 item 4), related to HF-03	GOV	added before the first run; until then final-status rows are decided with a fresh live sign-in (HF-10)

ID	Decision	Seat	Default
H-05	The engine register at the tool gateway: which families it carries; each engine's confidentiality ceiling, retention and use (change-gateway-policy , Standard §4.4 item 11)	GOV and the TM Cell, both signing	at least four families: at least two that may author and at least two used for reviewing only; ceilings such that two author engines of two families, and three reviewer engines of two other families, may receive the organisation's highest confidentiality level
H-06	An earlier, partial run before every stage is built	GOV	no: every stage is built before the first real run
H-07	Take records of the work as it runs today as an input: exports by role-type and process, never about a person, each admitted only as H-40 sets out (Annex H.8). The label observed is not used: this AIOC never watches work	GOV	directed (decisions/DR-0006): the input exists; where no export is admitted, the activity profile reads no observation and the run goes on
H-08	Whether any route other than the Chief of Staff of [ORG]AIOC-SL may ever ask the organisation. At 49GAION, the matching line on Protocol §13 item 7: if a part ever replaces Source Lifecycle, the Protocol must name the new asking route	GOV (the organisation's; and 49GAION's governing seat for the Protocol)	no: adapters only read
H-09	Send the drafts of upstream/ to 49GAION; each reaches 49GAION only as a pull request merged by 49GAION's governing seat	GOV	after v0.4.0 is accepted

ID	Decision	Seat	Default
H-10	Install [ORG]AIOC-TOM (create-or-install-unit); its change-parent; its restart-stopped-unit	TM Cell	none
H-11	Who holds the TOM Design Lead position, and how, is decided outside the system (49GAION/DR-0048, DR-0049). Nothing of it enters this blueprint	the organisation's own mechanism, outside the system	none: the AIOC does not join
H-12	reduced_review_permitted	TM Cell	no
H-13	review_budget_per_run, and the AIOC's budget_cap (widen-ai-budget-or-unattended-action)	TM Cell	none: must be set
H-14	non_response_passes and the waiting deadlines	TM Cell	none: must be set
H-15	Accept the Source Lifecycle additions SL-A1 to SL-A13, with SL-A8 as amended, as a change to [ORG]AIOC-SL proposed through OCO	TM Cell	required before joining (T0: SL contract not supported)
H-16	Create <org>aioc-tom-records and <org>aioc-tom-records-restricted, recorded first (Naming Standard R9)	TM Cell	before the first run
H-17	The first real run while the AIOC's card status is DRAFT (path qualification and the batteries not yet passed)	TM Cell	yes, once the joining test and every seat test pass; the walkthrough's clean-report wording applies

ID	Decision	Seat	Default
H-18	A change to a fixed test of TEST-TOM or its threshold, including the release lint deny-list (change-fixed-test-or-threshold, HF-14)	TM Cell (owner of TEST-TOM); the Design Lead for per-part tests	none
H-19	Approve the Design Request in Source Lifecycle, and open the run	Requesting Authority	none: no run
H-20	Frame rows (S1): scope, exclusions, the entity roster (committed or context), contradictions touching them; and the delegable confirmation rows; and, where the two frame drafts read the request differently and the sources do not settle it, which reading is meant (reading A, reading B, both readings)	Requesting Authority	undecided
H-21	Structure rows (S2): each child; the first slice	Requesting Authority	undecided
H-22	Child brief rows (S3): each child's criteria; contradictions touching the child	Requesting Authority	undecided
H-23	Seam rows (S6) that cannot be closed upstream	Requesting Authority	undecided
H-24	The choice on each material finding notice (SUPPLY, WITHDRAW, REDESIGN, ACCEPT AS KNOWN DEFECT, CLOSE)	Requesting Authority	none adopted; the step stays held

ID	Decision	Seat	Default
H-25	Each element row (ACCEPT, CORRECT, REJECT) and the final status; each entry of the role register is an element	Requesting Authority	undecided
H-26	Withdraw a run, or close one held on a revoked connection	Requesting Authority	the run stays held
H-27	Open a continuation run (the next slice), naming the children to change	Requesting Authority	none
H-28	The connection grant to Source Lifecycle (connect-external-port), its widening (grant-or-widen-access) and its revocation	the [ORG]HOAC-ACS seat over [ORG]AIOC-SL	none: T0 fails
H-29	Inside a run: acknowledge the run preview and accept a reduced level within H-12; accept single authoring at S5 within H-32; rule on contested MATERIAL findings; decide a re-base; approve a second run in parallel; attest T0–T7; set the slots it owns (§15); sign the listed types of its authority_scope on this AIOC's Cells	the TOM Design Lead ([ORG]HOAC-ACS of this AIOC)	none taken; the step waits
H-30	The run depth (light, standard, deep), stated in the Design Request (DRQ-01 field 9)	Requesting Authority	standard
H-31	run_depth_permitted: the deepest run depth a run may take	TM Cell	standard

ID	Decision	Seat	Default
H-32	single_author_permitted: whether S5 may run with one author where two would lower its review	TM Cell	no
H-33	Role rows (S6): whether two role names are one role	Requesting Authority	undecided
H-34	Take a version of the 49GAION reference-model catalogue into use: pin it in contracts/PINS.json	TM Cell (made through OCO)	none: the catalogue is declared unavailable
H-35	The connections with [ORG]AIOC-PPD (release notices out; returns in, and their answers) and with [ORG]AIOC-DI (returns in, and their answers) (connect-external-port), their widening (grant-or-widen-access) and their revocation	the [ORG]HOAC-ACS position over this AIOC (the TOM Design Lead's)	none: no notice is sent and no return is received; this AIOC stands alone
H-36	return_round_deadline and return_rounds_max (§15)	TM Cell	none: must be set; until set, every admitted return goes straight to the Requesting Authority with the two rows of §5.10, limit 6
H-37	Each return row (§5.10): open a continuation run naming the children to change; answer by decision; or set the return aside and carry its elements forward as a known defect or a decided assumption	Requesting Authority	undecided; past its deadline, escalated (§5.10, limit 6)

ID	Decision	Seat	Default
H-38	Classify each ended return: a defect of this AIOC, or normal discovery (§5.10, limit 8)	the TOM Design Lead	none taken; the classification waits, shown on the Panel
H-39	The strategy frame row (S1); the value case row and each element's priority (S8); a maturity target or a value-case target set by correction (Annex H.1, H.4, H.6, H.7)	Requesting Authority; purpose and values, the Cell carrying [ORG]HOAC-GOV (H-03)	undecided; a missing strategy element is not given, a missing target not set
H-40	Admit each export of records of the work in Source Lifecycle: the organisation's projection by role-type and process, removing who did the work; its impact assessment; its decision on it (HF-17; where the projection moves personal data, the listed type export-share-or-purge-personal-data) (Annex H.8; SL-A13)	the seat that holds data protection (HF-17), in the organisation's own mechanism	none: no export is admitted, and the activity profile reads no observation

H-01 to H-27, H-30 to H-34, H-36, H-37 and H-39 are taken by GOV, the TM Cell or the Requesting Authority. H-40 is taken in the organisation's own mechanism, outside this AIOC. H-28, H-29, H-35, H-38 and H-40 are listed so that the list is complete; they are not design questions.

Money: the AI budget cap and the review budget (H-13); the deepest run depth permitted (H-31). **People:** none; this AIOC writes no position. **Legal:** none decided here; the perimeter list goes to a human legal reviewer.



Section 14

Change log

Blueprint version	Date	Change	Template version	Commit
v0.1.0	2026-10-14	First draft: six earlier capabilities joined as Pods; documents only; needs requests to the Chief of Staff of [ORG]AIOC-SL; TOM Release filed into Source Lifecycle	v1.5.0	not published
v0.2.0	2026-10-14	Applies three independent reviews (fidelity, conformance, direction): the AIOC answers to the TM Cell; the run envelope and the Design Request are defined; the run has states; the order's audits come before its decision rows; decomposition comes before the parent floor; Gate A, admissible criteria and decision rules are written out; confirmation never lowers a label; the source reference matches what Source Lifecycle serves; listed types need a gateway signature; the Missing Report; the material finding notice; reviewer independence; per-part tests and bot identity cards; organisation-specific history moved to the private code project	v1.5.0	this commit
v0.3.0	2026-10-15	Closes the blueprint. Applies four independent reviews of v0.2.0 and their debate, under the directions recorded in decisions/DR-0002. Changes: a one-page front and a closure statement; nine stages, each with one input, one output and one done-test (§5.0); the Input Pack, versioned, with atomising and labelling in the socket; adapters, one file per source type; the Review Panel rule (three to five engines from two or more families at every review point; a blind pass; a debate held to quotes; settlement by code; rulings by the Design Lead on contested findings only); Hand-offs in place of Group Assembly, with no model of the whole; a release that is fully about the operating model (quality checks in place of review records; the release lint); carry-forward of earlier slices; the restored question set, request kinds, lens pass, understanding check and Gate B rows; Design Brief and Readiness Report; one list of human decisions; card lifecycle_state: designed; TEST-TOM v2	v1.5.0	this commit

Blueprint version	Date	Change	Template version	Commit
v0.4.1	2026-10-15	Wording only, under the direction recorded in decisions/DR-0004: every human position is a HOAC position, a legal persona; no natural person exists in the system (Standard §1.1, item 9, proposed by 49GAION/DR-0013). The Design Lead position in §2.1 and §3; a §3 note; SL-A2 in docs/INTAKE-WITH-SOURCE-LIFECYCLE.md addresses a position, not a holder. No rule, stage, test or card changes	v1.5.0	this commit
v0.4.0	2026-10-15	Adds four features, under the direction recorded in decisions/DR-0003, and removes or weakens nothing of v0.3.0: the role register, made by code at S6 (§5.6; Annex E.17); the run depth (Annex F); two independent authors at S1 and S5, with the Second Request Analyst, the Frame Reconciler and the Second Designer (Annex G); reference models from the 49GAION Library, as methods never evidence (§5.5; Annex B.5; Annex E.19). TEST-TOM v3 (cases 85–99; TDL-7)	v1.5.0	this commit
v0.5.0	2026-10-16	Links this AIOC directly to the stages after it, under the direction recorded in decisions/DR-0005 and 49GAION/DR-0016; removes nothing. A release notice to Process and Position Design on every filing (§5.9 step 8a; contracts/release-notice.schema.json); the returns of Process and Position Design and of Delivery and Implementation received directly, under the Library's design return contract pinned in contracts/PINS.json, and carried round by round to one of four ends (§5.10, new; Annex E.20 to E.22; slots return_round_deadline and return_rounds_max; H-35 to H-38); the mother is the Library part (§0). Wording, under 49GAION/DR-0015: two positions are two parties, and no check reads who occupies a position (T0 condition (d), H-02, H-11, the filing); no text states a number of people, even to deny one. TEST-TOM v4 (cases 100–108; TDL-8; cases 18, 27, 29 and 86 reworded)	v1.5.0	this commit

Blueprint version	Date	Change	Template version	Commit
v0.6.0	2026-10-16	Adds what the model also states and reads, under the direction recorded in decisions/DR-0006; removes nothing (Annex H, new): the strategy frame at S1, asked once through EX-10 and decided in one row (H.1); each capability's delivery model, make-or-buy and maturity, and each value stream's criticality, at S5 (H.2–H.4); each outside party's journey at S6 (H.5); the value case and each element's priority, by code, at S8 (H.6, H.7); records of the work as it runs today, exports by role-type and process only, the refusal person-level records and the activity profile (H.8); the organisation dictionary read where served (H.9). Four release parts (twenty in all); a decided priority on each element decision; the v0.6 contracts (tom-release, input-pack, needs-request, needs-answer); adapter ADP-SL v1.1.0; H-07 rewritten; H-39, H-40. TEST-TOM v5 (cases 109–123)	v1.5.0	this commit
v0.6.1	2026-10-19	The Chief of Staff takes its COS code: [ORG]CELL-TCS → [ORG]CELL-COSTOM (Naming Standard R16, 49GAION/DR-0028; decisions/DR-0007). The old code is retired and kept in the card's lineage; the part test is renamed; the needs-request contract names the new code and takes the id v0.6.1; nothing else changes	v1.5.0	this repository's pull request for this version
v0.6.2	2026-10-29	Released on the governing seat's word (decisions/DR-0010): it carries the records since v0.6.1, decisions/DR-0008, decisions/DR-0009; positions only (49GAION/DR-0048, DR-0049). Every card takes v0.6.2; the Protocol pin moves to the Operating Protocol v2.1.0. No other rule changes.		

Blueprint version	Date	Change	Template version	Commit
v0.6.3	2026-10-29	Corrections under 49GAION/DR-0052 (decisions/DR-0011): installation follows the current roles (the core designs and proposes, the governing seat approves, the Top Management Group installs), the withdrawn AION Builder gone; positions in place of persons (the TOM Design Lead, the position that can answer); the gateway runs as code beside the core. Every card takes v0.6.3. The Protocol pin moves to the Operating Protocol v2.2.0. No rule of this part is added or dropped.		



Section 15

Configuration

The template's slots, all none: must be set until the organisation sets them, plus these slots of this AIOC:

Slot	Card key	Default offered	Allowed values	Set by	New or changed
Requesting Authority Cell	requesting_authority_cell	none: must be set (§2.1)	the Cell carrying [ORG]HOAC-GOV, or a human Cell it delegated the design decisions to	the Cell carrying [ORG]HOAC-GOV	setter fixed; card key new
Rows per needs pass	rows_per_pass	12 (an explainer counts as three rows)	1–12	TOM Design Lead	card key new
Passes past their waiting deadline before a non-response stop	non_response_passes	none: must be set	≥ 1	the Cell carrying [ORG]HOAC-TM	setter fixed; card key new
Most decision passes per release	decision_passes_max	3	≥ 1	TOM Design Lead	card key new
Explainers per pass	explainers_per_pass	2	0–4	TOM Design Lead	card key new

Slot	Card key	Default offered	Allowed values	Set by	New or changed
Internal round caps	round_caps	Request Reconstruction 12; Structuring 8; Entity Model Design 8	may be set lower, never higher	TOM Design Lead	renamed; card key new
Repair cycles before a hold	repair_cycles	2	1–2	TOM Design Lead	card key new
Situation mix	situation_mix	standard library ≥ 60%, blind pack ≥ 30%	may be raised, never lowered	TOM Design Lead	card key new
Runs in parallel per organisation	parallel_runs	1	1; a second run needs the TOM Design Lead's approval, recorded per run	—	card key new
Engines per review point	review_engines	3; at the run depth deep, 5 for the run (Annex F)	3–5	TOM Design Lead, within the run's review budget	**new** (replaces the single reviewer slot of v0.2.0)
Families per review	review_families_min	2 (the draw takes 3 where the register allows)	2–3	TOM Design Lead	**new**
Debate rounds	—	2	fixed by §5.8 rule 12; not a slot	—	listed so that no one adds a slot

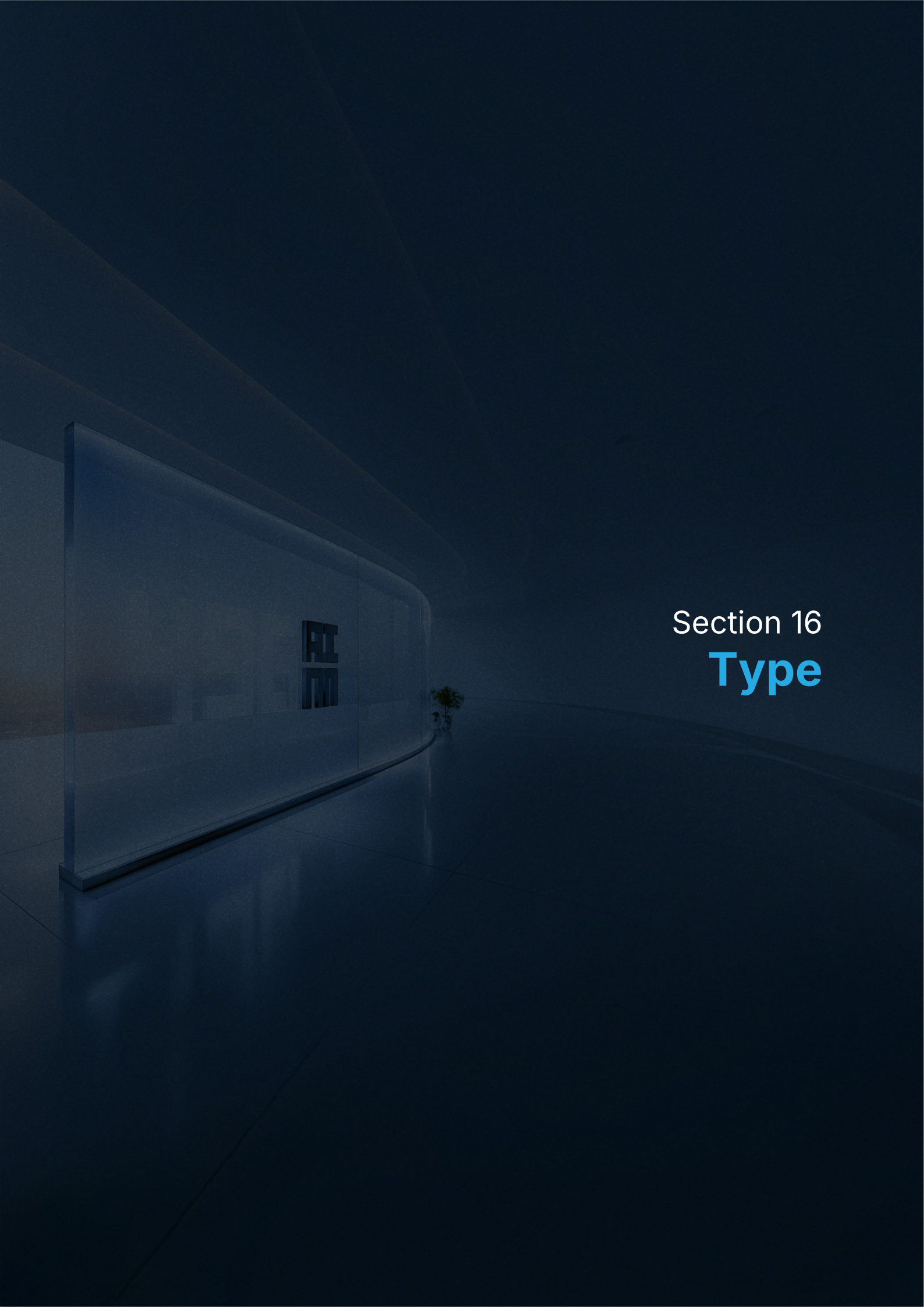
Slot	Card key	Default offered	Allowed values	Set by	New or changed
Reduced review permitted	reduced_review_permitted	no	yes, no	**the Cell carrying [ORG]HOAC-TM** (never the [ORG]HOAC-ACS seat of this AIOC, which is the Design Lead: no seat lowers the review of its own unit, Standard §2.3 item 9)	**new**
Review budget per run	review_budget_per_run	none: must be set	in the gateway's metered unit, within budget_cap	the Cell carrying [ORG]HOAC-TM	**new**
Lazy-atomising threshold	lazy_atomising_threshold	1,000 held sources in scope	≥ 1	TOM Design Lead	**new**. At or below it, every held source in scope is atomised at P0. Above it, the Design Request and the sources the read plan opens are atomised, each opening as a new pack version (made_because: source opened).
Lenses per frame	lenses_max	6; it is the cap at standard; light caps at 3 and deep at 8 (Annex F); never fewer than 2	2–8	TOM Design Lead	**new**

Slot	Card key	Default offered	Allowed values	Set by	New or changed
Run depth permitted	run_depth_permitted	standard	light, standard, deep	the Cell carrying [ORG]HOAC-TM (cost)	**new** (v0.4.0). The run's own depth is stated by the Requesting Authority in the Design Request (DRQ-01 field 9; Annex F), never by a slot
Return round deadline	return_round_deadline	none: must be set	a period	the Cell carrying [ORG]HOAC-TM	**new** (v0.5.0); §5.10, limit 5
Most rounds of a return	return_rounds_max	none: must be set	≥ 1	the Cell carrying [ORG]HOAC-TM	**new** (v0.5.0); §5.10, limit 6
Single authoring permitted	single_author_permitted	no	yes, no	the Cell carrying [ORG]HOAC-TM (never the [ORG]HOAC-ACS seat of this AIOC)	**new** (v0.4.0); Annex G.7
Request kind profiles	—	the list of docs/ASKING-CATALOGUE.md §3b	added by a tested change	proposed by the TOM Design Lead, made through OCO	**new**; configuration in the catalogue
Adapter register	—	adapters/README.md: ADP-SL	added by a tested change	made through OCO; each connection signed by the seat over that source	**new**

Slot	Card key	Default offered	Allowed values	Set by	New or changed
Engine register	—	held at the tool gateway (Annex E.12)	—	change-gateway-policy, signed by the Cell carrying [ORG]HOAC-TM and the Cell carrying [ORG]HOAC-GOV	**new** (read by this AIOC); its default changed in v0.4.0 (H-05: four families)
Release lint deny-list	—	tests/release-lint/DENY-LIST.json v1	a word is added by a tested change; a removal is change-fixed-test-or-threshold	the Cell carrying [ORG]HOAC-TM, owner of TEST-TOM	**new**
Reference-model catalogue pin	—	contracts/PINS.json: "not in the Library" (the catalogue is declared unavailable)	a version, commit and SHA-256 of the 49GAION catalogue	the Cell carrying [ORG]HOAC-TM, made through OCO as a tested change (H-34)	**new**
Template slots	office_accountable_human, central_head, discussion_limits, waiting_deadlines, bounce_limit, chain_depth (at least 1; needs requests are sub-items of depth 1), halt_request_priority (immediate), budget_cap, unattended_action (read-record-triage)	as the template	—	as the template; budget_cap and waiting_deadlines by the Cell carrying [ORG]HOAC-TM	unchanged values

Slot	Card key	Default offered	Allowed values	Set by	New or changed
~~the single reviewer slot of v0.2.0~~	—	—	—	—	**deleted**

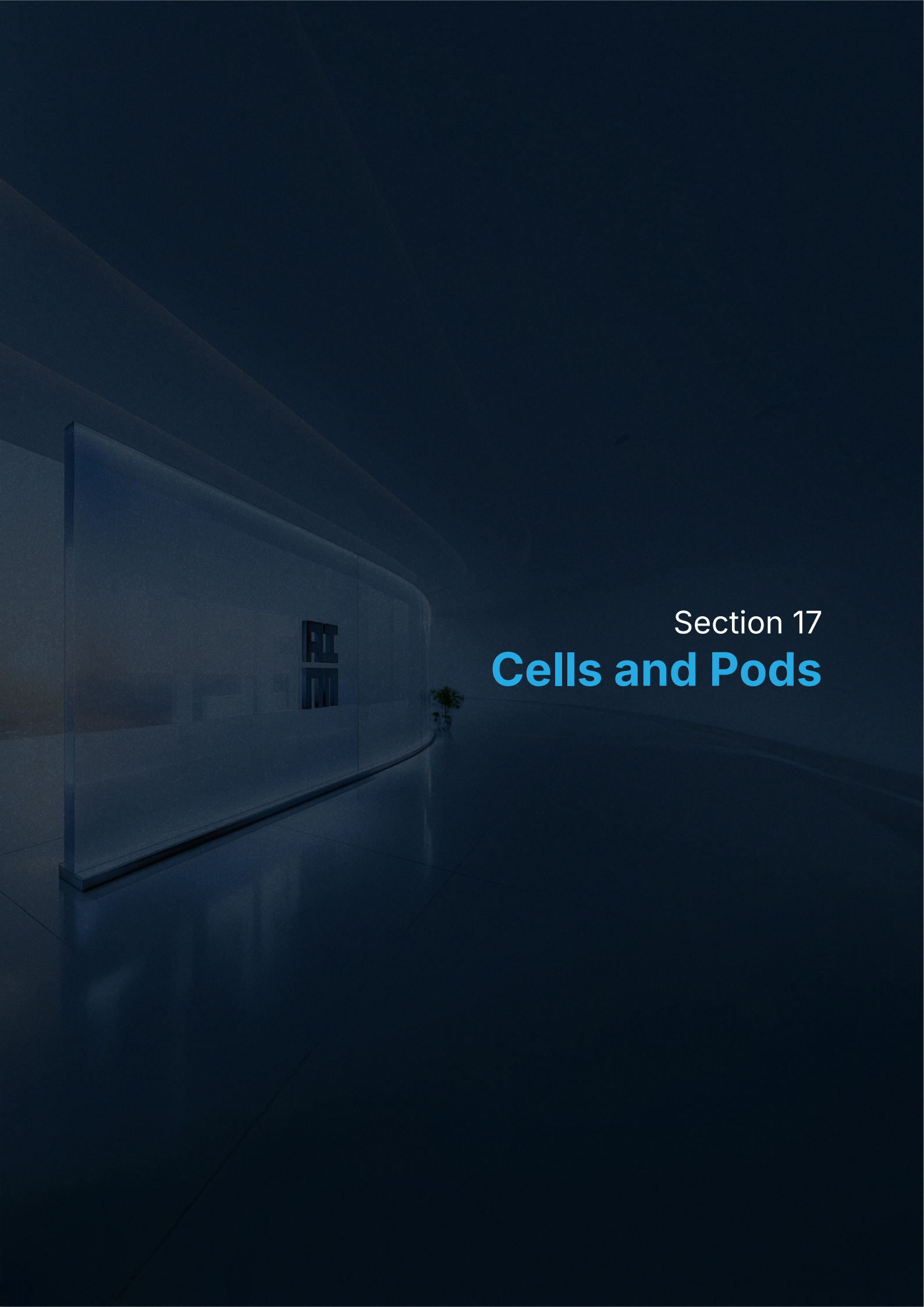
Waiting deadlines and escalation follow the template slot waiting_deadlines (Protocol §4.6); re-sending follows bounce_limit (Protocol §4.4).



Section 16

Type

- **Type:** Specialised.
- **Why this AIOC exists:** an organisation needs a target operating model that is honest about its evidence and decided by its own authority; this AIOC makes one from the organisation's documents alone.
- **Type-specific test:** TEST-TOM v5, case 27 · result NOT_RUN.



Section 17

Cells and Pods

The parts inside this AIOC (docs/CELLS-AND-PODS.md of 49GAION-aioc). The role and nature of this AIOC are read from the type of its top Cell, the TOM Design Lead.

Code	Kind	Type	Filler kind	HOAC label	Parent	Attached to	Fixed test (owner)	External ports
[ORG]CELL-TDL	CELL	Human Seat	human	[ORG]HOAC-ACS	[ORG]AIOC-TOM	none	TEST-TOM-TDL v5 (the Cell carrying [ORG]HOAC-TM)	none
[ORG]CELL-COSTOM	CELL	AI Chief of Staff	ai	none	[ORG]AIOC-TOM	none	TEST-TOM-COSTOMv5 ([ORG]CELL-TDL)	[ORG]AIOC-SL: sl-in and sl-out; [ORG]AIOC-PPD: ppd-in and ppd-out; [ORG]AIOC-DI: di-in and di-out; all through the tool gateway
[ORG]POD-TRR	POD	AI Pod	—	none	[ORG]AIOC-TOM	—	TEST-TOM-TRR v5 ([ORG]CELL-TDL)	—
[ORG]CELL-TSP	CELL	AI Agent	ai	none	[ORG]POD-TRR	none	TEST-TOM-TSP v5 ([ORG]CELL-TDL)	none
[ORG]CELL-TRA	CELL	AI Agent	ai	none	[ORG]POD-TRR	none	TEST-TOM-TRA v5 ([ORG]CELL-TDL)	none
[ORG]CELL-TSRA	CELL	AI Agent	ai	none	[ORG]POD-TRR	none	TEST-TOM-TSRA v5 ([ORG]CELL-TDL)	none
[ORG]CELL-TFR	CELL	AI Agent	ai	none	[ORG]POD-TRR	none	TEST-TOM-TFR v5 ([ORG]CELL-TDL)	none

Target Operating Model

Code	Kind	Type	Filler kind	HOAC label	Parent	Attached to	Fixed test (owner)	External ports
[ORG]POD-TS	POD	AI Pod	—	none	[ORG] AIOC-TOM	—	TEST-TOM-TS v5 ([ORG]CELL-TDL)	—
[ORG]CELL-TWS	CELL	AI Agent	ai	none	[ORG] POD-TS	none	TEST-TOM-TWS v5 ([ORG]CELL-TDL)	none
[ORG]POD-TEP	POD	AI Pod	—	none	[ORG] AIOC-TOM	—	TEST-TOM-TEP v5 ([ORG]CELL-TDL)	—
[ORG]CELL-TP	CELL	AI Agent	ai	none	[ORG] POD-TEP	none	TEST-TOM-TP v5 ([ORG]CELL-TDL)	none
[ORG]POD-TEM	POD	AI Pod	—	none	[ORG] AIOC-TOM	—	TEST-TOM-TEM v5 ([ORG]CELL-TDL)	—
[ORG]CELL-TD	CELL	AI Agent	ai	none	[ORG] POD-TEM	none	TEST-TOM-TD v5 ([ORG]CELL-TDL)	none
[ORG]CELL-TLR	CELL	AI Agent	ai	none	[ORG] POD-TEM	none	TEST-TOM-TLR v5 ([ORG]CELL-TDL)	none
[ORG]CELL-TSD	CELL	AI Agent	ai	none	[ORG] POD-TEM	none	TEST-TOM-TSD v5 ([ORG]CELL-TDL)	none
[ORG]POD-TH	POD	AI Pod	—	none	[ORG] AIOC-TOM	—	TEST-TOM-TH v5 ([ORG]CELL-TDL)	—
[ORG]CELL-THA	CELL	AI Agent	ai	none	[ORG] POD-TH	none	TEST-TOM-THA v5 ([ORG]CELL-TDL)	none

Code	Kind	Type	Filler kind	HOAC label	Parent	Attached to	Fixed test (owner)	External ports
[ORG]POD-TW	POD	AI Pod	—	none	[ORG] AIOC-TOM	—	TEST-TOM-TW v5 ([ORG]CELL-TDL)	—
[ORG]CELL-TSA	CELL	AI Agent	ai	none	[ORG] POD-TW	none	TEST-TOM-TSA v5 ([ORG]CELL-TDL)	none
[ORG]CELL-TSW	CELL	AI Agent	ai	none	[ORG] POD-TW	none	TEST-TOM-TSW v5 ([ORG]CELL-TDL)	none
[ORG]POD-TR	POD	AI Pod	—	none	[ORG] AIOC-TOM	—	TEST-TOM-TR v5 ([ORG]CELL-TDL)	—
[ORG]CELL-TBRA	CELL	AI Agent	ai	none	[ORG] POD-TR	none	TEST-TOM-TBRA v5 ([ORG]CELL-TDL)	none
[ORG]CELL-TBRB	CELL	AI Agent	ai	none	[ORG] POD-TR	none	TEST-TOM-TBRB v5 ([ORG]CELL-TDL)	none
[ORG]CELL-TBRC	CELL	AI Agent	ai	none	[ORG] POD-TR	none	TEST-TOM-TBRC v5 ([ORG]CELL-TDL)	none
[ORG]CELL-TBRD	CELL	AI Agent	ai	none	[ORG] POD-TR	none	TEST-TOM-TBRD v5 ([ORG]CELL-TDL)	none
[ORG]CELL-TBRE	CELL	AI Agent	ai	none	[ORG] POD-TR	none	TEST-TOM-TBRE v5 ([ORG]CELL-TDL)	none

The card data (the blueprint governs). The block below holds every value of every Pod and Cell card and of the bot identity card of every AI Cell. tools/make_cards.py expands it into the complete forms of the template (§17 of

the template; spec/schemas/unit-card.schema.json and bot-identity-card.schema.json of 49GAION-protocol), writes them into cards/ as generated copies, and checks them: against the schemas; against the code rules of tests/cards/check_cards.py (a fixed test is never owned by the part itself; a bot never claims a right to sign; no credential is shared); and against this blueprint (the inner units of the AIOC card are exactly the parts whose parent is the AIOC, and the AIOC is accountable to no Cell inside it). Each port gets the standard fields (identification, enforcement at the shared gateway outside the unit, audit_trail, required_tests); the stop door accepts only stop.

```
parts:
  defaults:
    accountable_to: "[ORG]CELL-TDL"
    card_version: "v0.6.3"
    version: "0.6.3"
    lifecycle_state: "designed"
    test_version: "v5"
    test_owner: "[ORG]CELL-TDL"
    audit_trail: "the run record of [ORG]AIOC-TOM"
    hu_in: ["run control and configuration from the TOM Design Lead (never a source
)"]
    hu_out: ["run report to the TOM Design Lead"]
  cells:
    - code: "[ORG]CELL-TDL"
      full_name: "TOM Design Lead"
      parent: "[ORG]AIOC-TOM"
      type: "Human Seat"
      filler_kind: "human"
      hoac: "[ORG]HOAC-ACS"
      accountable_to: "[ORG]CELL-TMS"       # example code of the Cell carrying [OR
G]HOAC-TM
      entity: "<entity>"
      authority_scope:
        decision_types: ["change-fixed-test-or-threshold", "create-or-install-unit"
, "change-filler", "retire-unit", "restart-stopped-unit", "move-work-toward-ai"]
```

```
        entity: "<entity>"
        expires: "2027-10-14"               # placeholder; each organisation sets
it
        authority: "delegated"
      test: { id: "TEST-TOM-TDL", cases: "TDL-1 to TDL-8", owner: "[ORG]CELL-TMS" }
      ai_in: ["run brief", "run preview", "ruling sheets for contested MATERIAL fin
dings", "release notice", "ended returns to classify"]
      ai_out: ["run control and configuration"]
      hu_in: ["direction from the Cell carrying [ORG]HOAC-TM"]
      hu_out: ["rulings, attestations, reduced-level acceptances, re-base decisions
 and the classification of ended returns, under live sign-in", "report to the Cell
carrying [ORG]HOAC-TM"]
    - code: "[ORG]CELL-COSTOM"
      code_rule: "R16"                      # 49GAION/DR-0028; replaces [ORG]CELL-TC
S (retired, R5)
      lineage: { id: "[ORG]CELL-TCS", version: "0.6.0", relation: "derived" }
      full_name: "TOM Chief of Staff"
      role: "Chief of Staff"
      parent: "[ORG]AIOC-TOM"
      type: "AI Chief of Staff"
      filler_kind: "ai"
      cell_kind: "chief-of-staff-cell"
      test: { id: "TEST-TOM-COSTOM", cases: "1-13, 23-36, 40, 43, 44, 46, 49-52, 61
, 64, 67-69, 75, 76, 78, 80-82, 85, 87, 89, 90, 92, 94-96, 98-108" }
      ai_in: ["source reference", "needs answer", "needs status", "needs candidate"
, "locked outputs of the Pods", "settled review records", "return"]
      ai_out: ["Input Pack version", "dispatch to a Pod", "review packet", "needs r
equest", "missing report", "needs status", "TOM Release filing", "difference regist
er", "role register", "release notice", "return brief", "return answer"]
      hu_out: ["run brief to the TOM Design Lead", "run preview to the TOM Design L
ead", "ruling sheets to the TOM Design Lead", "release notice to the Requesting Aut
hority's Cell"]
      external_ports:
        - { port: "sl-in", direction: "input", accepted_content: ["source reference
", "needs answer", "needs status"], authority_limit: "the read scope signed under c
onnect-external-port", audit_trail: "unit register connected and disconnected event
s; serve log of [ORG]AIOC-SL" }
```

```
        - { port: "sl-out", direction: "output", accepted_content: ["needs request"
, "missing report", "needs status", "TOM Release filing"], authority_limit: "the wr
ite and request scope signed under connect-external-port: sources of origin kind op
erating model or design report only", audit_trail: "unit register connected and dis
connected events; serve log of [ORG]AIOC-SL" }

        - { port: "ppd-out", direction: "output", accepted_content: ["release notic
e", "return answer"], authority_limit: "notices and answers by reference only, unde
r the scope signed under connect-external-port over this AIOC", audit_trail: "unit
register; the Work Item record" }

        - { port: "ppd-in", direction: "input", accepted_content: ["return"], autho
rity_limit: "returns only, under the scope signed under connect-external-port over
this AIOC", audit_trail: "unit register; the Work Item record" }

        - { port: "di-out", direction: "output", accepted_content: ["return answer"
], authority_limit: "answers by reference only, under the scope signed under connec
t-external-port over this AIOC", audit_trail: "unit register; the Work Item record"
 }

        - { port: "di-in", direction: "input", accepted_content: ["return"], author
ity_limit: "returns only, under the scope signed under connect-external-port over t
his AIOC", audit_trail: "unit register; the Work Item record" }

      bot:

        allowed_actions: ["dispatch Cells of this AIOC", "build Input Pack versions
 through the code steps", "lock outputs and run code steps", "draw review engines a
nd run the review code steps", "write the run record", "send needs requests, Missin
g Reports and status messages to [ORG]AIOC-SL", "send run previews and ruling sheet
s to the TOM Design Lead", "file a release whose final status is decided", "compare
 two locked drafts and compile the role register through the code steps", "send the
 release notice of a filed release to [ORG]AIOC-PPD", "admit or refuse a return by
the pinned contract and record it", "generate return briefs through the code step a
nd send their rows to [ORG]AIOC-SL", "send return answers to the sender of each ret
urn"]

        ask_first_actions: ["a second run in parallel", "a budget above the cap", "
a re-base of the Input Pack", "a reduced review level", "single authoring at S5", "
any change to configuration", "anything outside its role card"]

    - { code: "[ORG]CELL-TSP", full_name: "TOM Source Proxy", role: "Source Proxy",
 parent: "[ORG]POD-TRR", type: "AI Agent", filler_kind: "ai", test: { id: "TEST-TOM
-TSP", cases: "4, 5, 14, 92" }, ai_in: ["Input Pack atoms", "question from the Requ
est Analyst"], ai_out: ["cited answer or refusal"], bot: { allowed_actions: ["answe
r from the sources with a citation", "refuse"], ask_first_actions: ["anything outsi
de its role card"] } }

    - { code: "[ORG]CELL-TRA", full_name: "TOM Request Analyst", role: "Request Ana
lyst", parent: "[ORG]POD-TRR", type: "AI Agent", filler_kind: "ai", test: { id: "TE
ST-TOM-TRA", cases: "8, 14, 15, 28, 73-75, 83, 91, 92" }, ai_in: ["Input Pack atoms
", "cited answer or refusal", "decided structure"], ai_out: ["Design Brief draft",
"Gate A record", "party register", "lens record", "needs candidate"], bot: { allowe
```

d_actions: ["draft and label the Design Brief", "derive lenses and read the sources through them, with citations", "write the party register", "raise needs candidates with leverage proofs"], ask_first_actions: ["anything outside its role card"] } }

- { code: "[ORG]CELL-TSRA", full_name: "TOM Second Request Analyst", role: "Second Request Analyst", parent: "[ORG]POD-TRR", type: "AI Agent", filler_kind: "ai", test: { id: "TEST-TOM-TSRA", cases: "8, 14, 15, 73-75, 83, 91, 92" }, ai_in: ["Input Pack atoms", "cited answer or refusal"], ai_out: ["frame draft, written blind to the other author", "Gate A record", "party register", "lens record", "needs candidate"], bot: { allowed_actions: ["draft and label the frame blind to the other author's draft", "derive lenses and read the sources through them, with citations", "write a party register", "raise needs candidates with leverage proofs"], ask_first_actions: ["anything outside its role card"] } }

- { code: "[ORG]CELL-TFR", full_name: "TOM Frame Reconciler", role: "Frame Reconciler", parent: "[ORG]POD-TRR", type: "AI Agent", filler_kind: "ai", test: { id: "TEST-TOM-TFR", cases: "93, 94" }, ai_in: ["two locked frame drafts", "difference register", "the Input Pack atoms the drafts cite"], ai_out: ["reconciled frame", "the dispositions of the difference register"], bot: { allowed_actions: ["reconcile two locked frame drafts under blueprint Annex G", "record each difference's disposition with its reason and quotes"], ask_first_actions: ["anything outside its role card"] } }

- { code: "[ORG]CELL-TWS", full_name: "TOM Work Structurer", role: "Work Structurer", parent: "[ORG]POD-TS", type: "AI Agent", filler_kind: "ai", test: { id: "TEST-TOM-TWS", cases: "16, 42, 83" }, ai_in: ["decided Design Brief frame", "party register"], ai_out: ["Structure Proposal", "party register extension"], bot: { allowed_actions: ["propose children and checks", "record alternatives", "extend the party register with sibling children"], ask_first_actions: ["anything outside its role card"] } }

- { code: "[ORG]CELL-TP", full_name: "TOM Packager", role: "Packager", parent: "[ORG]POD-TEP", type: "AI Agent", filler_kind: "ai", test: { id: "TEST-TOM-TP", cases: "17, 41, 45" }, ai_in: ["markerised pack atoms", "class-only registry", "crossing candidates", "child row"], ai_out: ["packaging ledger"], bot: { allowed_actions: ["decide atom membership", "adjudicate crossing candidates with reasons"], ask_first_actions: ["anything outside its role card"] } }

- { code: "[ORG]CELL-TD", full_name: "TOM Designer", role: "Designer", parent: "[ORG]POD-TEM", type: "AI Agent", filler_kind: "ai", test: { id: "TEST-TOM-TD", cases: "18, 19, 38, 39, 91, 92, 97" }, ai_in: ["child package", "reference-model catalogue at the run's pinned version"], ai_out: ["design ledger, written blind to the other author"], bot: { allowed_actions: ["write a design ledger blind to the other author's ledger"], ask_first_actions: ["anything outside its role card"] } }

- { code: "[ORG]CELL-TLR", full_name: "TOM Ledger Reviser", role: "Ledger Reviser", parent: "[ORG]POD-TEM", type: "AI Agent", filler_kind: "ai", test: { id: "TEST-TOM-TLR", cases: "18, 93" }, ai_in: ["child package", "two locked drafts", "difference register", "review findings to repair"], ai_out: ["reconciled and revised ledger", "revision record", "the dispositions of the difference register"], bot: { allowed_actions: ["reconcile two locked design ledgers under blueprint Annex G", "revis

e a locked ledger in a fresh context"], ask_first_actions: ["anything outside its r
ole card"] } }

- { code: "[ORG]CELL-TSD", full_name: "TOM Second Designer", role: "Second Desi
gner", parent: "[ORG]POD-TEM", type: "AI Agent", filler_kind: "ai", test: { id: "TE
ST-TOM-TSD", cases: "18, 19, 38, 39, 91, 92, 97" }, ai_in: ["child package", "refer
ence-model catalogue at the run's pinned version"], ai_out: ["design ledger, writte
n blind to the other author"], bot: { allowed_actions: ["write a design ledger blin
d to the other author's ledger"], ask_first_actions: ["anything outside its role ca
rd"] } }

- { code: "[ORG]CELL-THA", full_name: "TOM Hand-off Analyst", role: "Hand-off A
nalyst", parent: "[ORG]POD-TH", type: "AI Agent", filler_kind: "ai", test: { id: "T
EST-TOM-THA", cases: "20, 37, 42, 65, 66, 70, 87" }, ai_in: ["design ledgers", "pai
ring table", "seam findings", "party register"], ai_out: ["hand-offs register", "co
unterparty register", "perimeter list", "deferred-boundary register"], bot: { allow
ed_actions: ["record each clean pair in the hand-offs register", "map counterpartie
s by class", "list the legal and regulatory perimeter", "register deferred boundary
 classes"], ask_first_actions: ["anything outside its role card"] } }

- { code: "[ORG]CELL-TSA", full_name: "TOM Situation Author", role: "Situation
Author", parent: "[ORG]POD-TW", type: "AI Agent", filler_kind: "ai", test: { id: "T
EST-TOM-TSA", cases: "21, 84, 91" }, ai_in: ["blind brief: entity type, accountabil
ity set and boundary list"], ai_out: ["blind situation pack"], bot: { allowed_actio
ns: ["write the blind situation pack"], ask_first_actions: ["anything outside its r
ole card"] } }

- { code: "[ORG]CELL-TSW", full_name: "TOM Situation Walker", role: "Situation
Walker", parent: "[ORG]POD-TW", type: "AI Agent", filler_kind: "ai", test: { id: "T
EST-TOM-TSW", cases: "21, 71" }, ai_in: ["sealed situation set", "operating model g
raph from the walk input"], ai_out: ["trace", "finding register"], bot: { allowed_a
ctions: ["walk sealed situations under the binding rules", "classify findings"], as
k_first_actions: ["anything outside its role card"] } }

- { code: "[ORG]CELL-TBRA", full_name: "TOM Blind Reviewer A", role: "Blind Rev
iewer A", parent: "[ORG]POD-TR", type: "AI Agent", filler_kind: "ai", test: { id: "
TEST-TOM-TBRA", cases: "22, 53, 55, 72 and the reviewer battery" }, ai_in: ["review
 packet", "collated findings of its review"], ai_out: ["review card", "positions on
 findings"], bot: { allowed_actions: ["write its own review card", "state its own p
ositions on the findings of its review"], ask_first_actions: ["anything outside its
 role card"] } }

- { code: "[ORG]CELL-TBRB", full_name: "TOM Blind Reviewer B", role: "Blind Rev
iewer B", parent: "[ORG]POD-TR", type: "AI Agent", filler_kind: "ai", test: { id: "
TEST-TOM-TBRB", cases: "22, 53, 55, 72 and the reviewer battery" }, ai_in: ["review
 packet", "collated findings of its review"], ai_out: ["review card", "positions on
 findings"], bot: { allowed_actions: ["write its own review card", "state its own p
ositions on the findings of its review"], ask_first_actions: ["anything outside its
 role card"] } }

- { code: "[ORG]CELL-TBRC", full_name: "TOM Blind Reviewer C", role: "Blind Rev
iewer C", parent: "[ORG]POD-TR", type: "AI Agent", filler_kind: "ai", test: { id: "

TEST-TOM-TBRC", cases: "22, 53, 55, 72 and the reviewer battery" }, ai_in: ["review packet", "collated findings of its review"], ai_out: ["review card", "positions on findings"], bot: { allowed_actions: ["write its own review card", "state its own positions on the findings of its review"], ask_first_actions: ["anything outside its role card"] } }

- { code: "[ORG]CELL-TBRD", full_name: "TOM Blind Reviewer D", role: "Blind Reviewer D", parent: "[ORG]POD-TR", type: "AI Agent", filler_kind: "ai", test: { id: "TEST-TOM-TBRD", cases: "22, 53, 55, 72 and the reviewer battery" }, ai_in: ["review packet", "collated findings of its review"], ai_out: ["review card", "positions on findings"], bot: { allowed_actions: ["write its own review card", "state its own positions on the findings of its review"], ask_first_actions: ["anything outside its role card"] } }

- { code: "[ORG]CELL-TBRE", full_name: "TOM Blind Reviewer E", role: "Blind Reviewer E", parent: "[ORG]POD-TR", type: "AI Agent", filler_kind: "ai", test: { id: "TEST-TOM-TBRE", cases: "22, 53, 55, 72 and the reviewer battery" }, ai_in: ["review packet", "collated findings of its review"], ai_out: ["review card", "positions on findings"], bot: { allowed_actions: ["write its own review card", "state its own positions on the findings of its review"], ask_first_actions: ["anything outside its role card"] } }

pods:

- { code: "[ORG]POD-TRR", full_name: "TOM Request Reconstruction", type: "AI Pod", inner_units: ["[ORG]CELL-TRA", "[ORG]CELL-TSP", "[ORG]CELL-TSRA", "[ORG]CELL-TFR"], top_cell: "[ORG]CELL-TRA", test: { id: "TEST-TOM-TRR", cases: "4, 5, 8, 14, 15, 28, 73-76, 83, 91-93" }, ai_in: ["Input Pack version", "needs answer", "decided structure"], ai_out: ["Design Brief", "party register", "needs candidate", "Readiness Report", "difference register"] }

- { code: "[ORG]POD-TS", full_name: "TOM Structuring", type: "AI Pod", inner_units: ["[ORG]CELL-TWS"], top_cell: "[ORG]CELL-TWS", test: { id: "TEST-TOM-TS", cases: "16, 42, 83" }, ai_in: ["decided Design Brief frame", "party register"], ai_out: ["Structure Proposal"] }

- { code: "[ORG]POD-TEP", full_name: "TOM Entity Packaging", type: "AI Pod", inner_units: ["[ORG]CELL-TP"], top_cell: "[ORG]CELL-TP", test: { id: "TEST-TOM-TEP", cases: "17, 41, 45" }, ai_in: ["decided child", "Input Pack atoms", "party register (code steps only)"], ai_out: ["child package"] }

- { code: "[ORG]POD-TEM", full_name: "TOM Entity Model", type: "AI Pod", inner_units: ["[ORG]CELL-TD", "[ORG]CELL-TLR", "[ORG]CELL-TSD"], top_cell: "[ORG]CELL-TD", test: { id: "TEST-TOM-TEM", cases: "18, 19, 38, 39, 91-93, 96, 97" }, ai_in: ["child package", "reference-model catalogue at the run's pinned version"], ai_out: ["design ledger", "TOM Blueprint", "Rationale and Evidence Brief"] }

- { code: "[ORG]POD-TH", full_name: "TOM Hand-offs", type: "AI Pod", inner_units: ["[ORG]CELL-THA"], top_cell: "[ORG]CELL-THA", test: { id: "TEST-TOM-TH", cases: "20, 37, 42, 65, 66, 70, 85-88" }, ai_in: ["design ledgers", "boundary registers", "party register"], ai_out: ["hand-offs register", "counterparty register", "perimeter list", "deferred-boundary register", "seam findings", "role register"] }

```
    - { code: "[ORG]POD-TW", full_name: "TOM Walkthrough", type: "AI Pod", inner_un
its: ["[ORG]CELL-TSW", "[ORG]CELL-TSA"], top_cell: "[ORG]CELL-TSW", test: { id: "TE
ST-TOM-TW", cases: "21, 71, 84, 91 and the walkthrough battery" }, ai_in: ["walk in
put", "blind brief"], ai_out: ["walkthrough report"] }

    - { code: "[ORG]POD-TR", full_name: "TOM Review", type: "AI Pod", inner_units:
["[ORG]CELL-TBRA", "[ORG]CELL-TBRB", "[ORG]CELL-TBRC", "[ORG]CELL-TBRD", "[ORG]CELL
-TBRE"], top_cell: "[ORG]CELL-TBRA", test: { id: "TEST-TOM-TR", cases: "22, 23, 41,
 53-64, 72, 88, 90, 91, 93, 94, 97 and the reviewer battery" }, ai_in: ["review pac
ket"], ai_out: ["review cards and positions"] }
```


- **Accountability.** Every Pod and Cell of this AIOC is accountable_to [ORG]CELL-TDL. The AIOC itself and the Design Lead are accountable_to the Cell carrying [ORG]HOAC-TM, which signs the AIOC's install, its change-parent and its restart-stopped-unit (Standard §5.3, item 4; §4.4, item 3).
- **The Design Lead's authority.** The authority_scope of [ORG]CELL-TDL lists the listed types it signs on this AIOC's Cells (Protocol §4.10, item 12). Every other listed type on this AIOC's parts is signed by the Cell carrying [ORG]HOAC-TM. The expiry date is a placeholder that each organisation sets.
- **Bots.** Each AI Cell's bot (eighteen) has its own identity card, with its own credential, never_signs: true and its memory sealed on a change of filler (Protocol §10, item 3); the cards are generated into cards/bots/. The credential IDs there are placeholders; the credentials themselves live only in the organisation's vault.
- **Engines.** A bot's identity card names no engine. The gateway binds engines per call, and the draw binds them per review (§8; §5.8).
- **Codes.** The codes follow the initials of each part's own name (Naming Standard R15), except the Chief of Staff's, [ORG]CELL-COSTOM, which takes COS and the initials of [ORG]AIOC-TOM (R16, 49GAION/DR-0028). All pass the collision check of §5 (table below). [ORG]POD-TOR, [ORG]POD-TGA, [ORG]CELL-TOA, [ORG]CELL-TA and [ORG]CELL-TPA were never registered; they are withdrawn and never issued again for any meaning. [ORG]CELL-TCS, the Chief of Staff's code until v0.6.0, is retired at v0.6.1 (R5) and never issued again.
- The Requesting Authority and the Chief of Staff of [ORG]AIOC-SL are not Cells of this AIOC.

Code checks (Naming Standard R15, R16 and §5), recorded on 15 October 2026; the Chief of Staff's COS code on 19 October 2026. Each organisation repeats the check of §5.2 against its own register before first use (R9, R15.3).

Code	Initials of the full name (R15)	§5.1: no AIOC, AION or 49GAION	§5.2: no match in this blueprint's codes	§5.3: not a version suffix or misleading acronym	§5.4: recorded before first use
[ORG]POD-TRR	**T**OM **R**equest **R**econstruction → TRR (3 letters, R15: 2–8)	pass	pass (no other TRR)	pass	blueprint §17, 15 October 2026; each organisation's register before first use
[ORG]CELL-TRA	**T**OM **R**equest **A**nalyst → TRA	pass	pass (distinct from [ORG]POD-TR and [ORG]CELL-TBRA)	pass	as above
[ORG]POD-TH	**T**OM **H**and-offs → TH (a hyphenated word counts once)	pass	pass	pass	as above
[ORG]CELL-THA	**T**OM **H**and-off **A**nalyst → THA	pass	pass	pass	as above
[ORG]CELL-TBRC	**T**OM **B**lind **R**eviewer **C** → TBRC	pass	pass	pass	as above
[ORG]CELL-TBRD	TOM Blind Reviewer D → TBRD	pass	pass	pass	as above
[ORG]CELL-TBRE	TOM Blind Reviewer E → TBRE	pass	pass	pass	as above

Code	Initials of the full name (R15)	§5.1: no AIOC, AION or 49GAION	§5.2: no match in this blueprint's codes	§5.3: not a version suffix or misleading acronym	§5.4: recorded before first use
[ORG]CELL-TSRA	**T**OM **S**econd **R**equest **A**nalyst → TSRA (4 letters)	pass	pass (distinct from [ORG]CELL-TSA, -TSP, -TRA)	pass	blueprint §17, 15 October 2026; each organisation's register before first use
[ORG]CELL-TFR	**T**OM **F**rame **R**econciler → TFR	pass	pass	pass	as above
[ORG]CELL-TSD	**T**OM **S**econd **D**esigner → TSD	pass	pass (distinct from [ORG]CELL-TD, -TSP, -TSA, -TSW)	pass	as above
[ORG]CELL-COSTOM	R16, item 2 (49GAION/DR-0028): COS and the initials of [ORG] AIOC-TOM → COSTOM. It replaces [ORG]CELL-TCS, retired at v0.6.1 (R5) and kept in the card's lineage	pass	pass (distinct from the COS codes of the other mothers: COSSL, COSOER, COSPC, COSDI, COSPPD, COSCH)	pass	blueprint §17, 19 October 2026; each organisation's register before first use

The codes checked against: [ORG]CELL-TDL, -TCS, -TSP, -TRA, -TWS, -TP, -TD, -TLR, -THA, -TSA, -TSW, -TBRA to -TBRE; [ORG]POD-TRR, -TS, -TEP, -TEM, -TH, -TW, -TR; the example codes [ORG]AIOC-CH, [ORG]AIOC-SL, [ORG]AIOC-OCO, [ORG]CELL-TMS, [ORG]CELL-GOV, [ORG]CELL-SLCS, [ORG]POD-RV; the withdrawn codes; and the three new codes against each other. On 19 October 2026, [ORG]CELL-COSTOM against all of these and against the COS codes of the other mothers.



Annex B

Labels and the honesty ceiling

B.1 Labels. Every material statement in a run carries exactly one label, and a standing. This AIOC assigns the label on admission, in the Input socket (§4.3), once per atom. It never reads it from a source reference.

Label	Meaning	Evidence it needs
source-backed	a source in Source Lifecycle says it	the source reference and anchor, and the source's status. A source that is draft, in review, approved but not yet in force, or served with no status caps the statement at inferred. A superseded source is cited only as history
inferred	derived from sources	what it was derived from, the reasoning, and the basis strength: single-source, multi-weak, multi-strong or contested-basis
organisation-stated	the organisation said it, in an approved explainer, a correction or a decision recorded in Source Lifecycle	the source reference and anchor of that statement
assumed	carried without evidence	what breaks if it is false, and when to re-check
unknown	not established	why the sources cannot answer it; whether it blocks

- **Order:** source-backed > inferred > organisation-stated > assumed > unknown. The label observed is not used: this AIOC never watches work. Records of the work as it runs today are read as exports, by role-type and process, never about a person (Annex H.8; §13, H-07).
- **Source Lifecycle's own labels** are kept statement by statement (adapters/source-lifecycle.adapter.md §5; B.1a). A statement Source Lifecycle labels stated is admitted as source-backed when it stands in a document, and as organisation-stated when it was recorded from conversation. inferred, assumed and unknown stay as they are. An approved

explainer never lifts a statement Source Lifecycle holds as inferred or assumed.

- **Confirmation never lowers a label, and never launders one.** An organisation's confirmation of an inferred item keeps the label inferred and adds the standing note confirmed → <confirmation reference>. The item is then released from the sticky-inferred rule; it never becomes source-backed. A correction replaces the item with the corrected text, labelled organisation-stated. "Don't know" leaves the item as it was, unconfirmed, with the answer recorded. No answer ever lowers a label.
- **No label laundering.** A label changes only on evidence of the destination's own class. A change of what a claim covers ("the source says X" becoming "X is true") needs its own record row even when the label stays.
- **Sticky inferred.** An inferred item keeps raising its question until it is confirmed, corrected or carried with its cost.
- **Standing:** uncontested, or contested → C-n (the contradiction row). A statement cannot read uncontested while an open row names it.

B.1a Reading by origin kind. It applies to every adapter.

Origin kind	Read as
document	a statement the source labels stated → source-backed, subject to the status caps
explainer	stated → organisation-stated; inferred, assumed and unknown stay as they are
confirmation	Annex B.1: a confirm keeps inferred and adds the standing note confirmed → <ref>; a correction's text → organisation-stated; "don't know" leaves the item as it was
status	not a claim about the organisation; it only sets the canonical status of its target source
export	source-backed as of its export date; the date (as_of) travels with every atom. An export of records of the work is admitted only by role-type and process (Annex H.8)

Origin kind	Read as
decision	the decided value is a decision, with the standing note decided → <ref>; a fact in a correction → organisation-stated
operating model (an earlier release)	its decided elements are carried as decided, with their original labels (label_basis: carried label); never upgraded
design report	history only; its claims keep their original labels and never become source-backed

B.2 Contradictions. A contradiction row is closed only as:

- decided by the Requesting Authority for this brief, as a non-delegable row (the losing side stays in the register, and the decision does not spread beyond this brief);
- decided with a cited precedence rule from a held source;
- or carried, with what it blocks.

Where both sides rest on the same source, a decision does not close the row; it is carried. A decision that a draft governs never upgrades a claim to source-backed. A contradiction touching ownership, a scope boundary or a success criterion is never turned into an assumption. A status record in Source Lifecycle says whether one source is in force; it never says which of two sources governs. Two live versions of one document, found at admission, reach the frame as candidate contradictions; they are never resolved at S0. Two sources that disagree are a contradiction whichever adapters served them.

B.3 Honesty ceiling. The ceiling is the weakest label carried by any load-bearing claim, named together with that claim. It is stated per committed entity, and for the present state, the outcome and each criterion, with the label distribution. The brief's ceiling is the lowest of its committed entities; a context entity's ceiling is stated separately. A claim is load-bearing by default if it is used in scope, an exclusion, an outcome, a constraint, a criterion, a child boundary, a needs request, the do-not-ask or do-not-use-as-fact list, or an open item. Placing it in an open item does not exempt it. Calling it not load-bearing needs a recorded reason and is a reviewable defect. No

committed entity sits in scope with ceiling unknown unless it is listed as a blocking unknown; a context entity's unknown creates no blocking unknown.

Pattern reliance. The ceiling also names, per child and for the release, each pattern that informed a load-bearing element (by its plain name and catalogue version), how many load-bearing elements it informed, and how many of those rest on no atom of the organisation. Such an element is assumed (B.5), so it sets the ceiling at assumed.

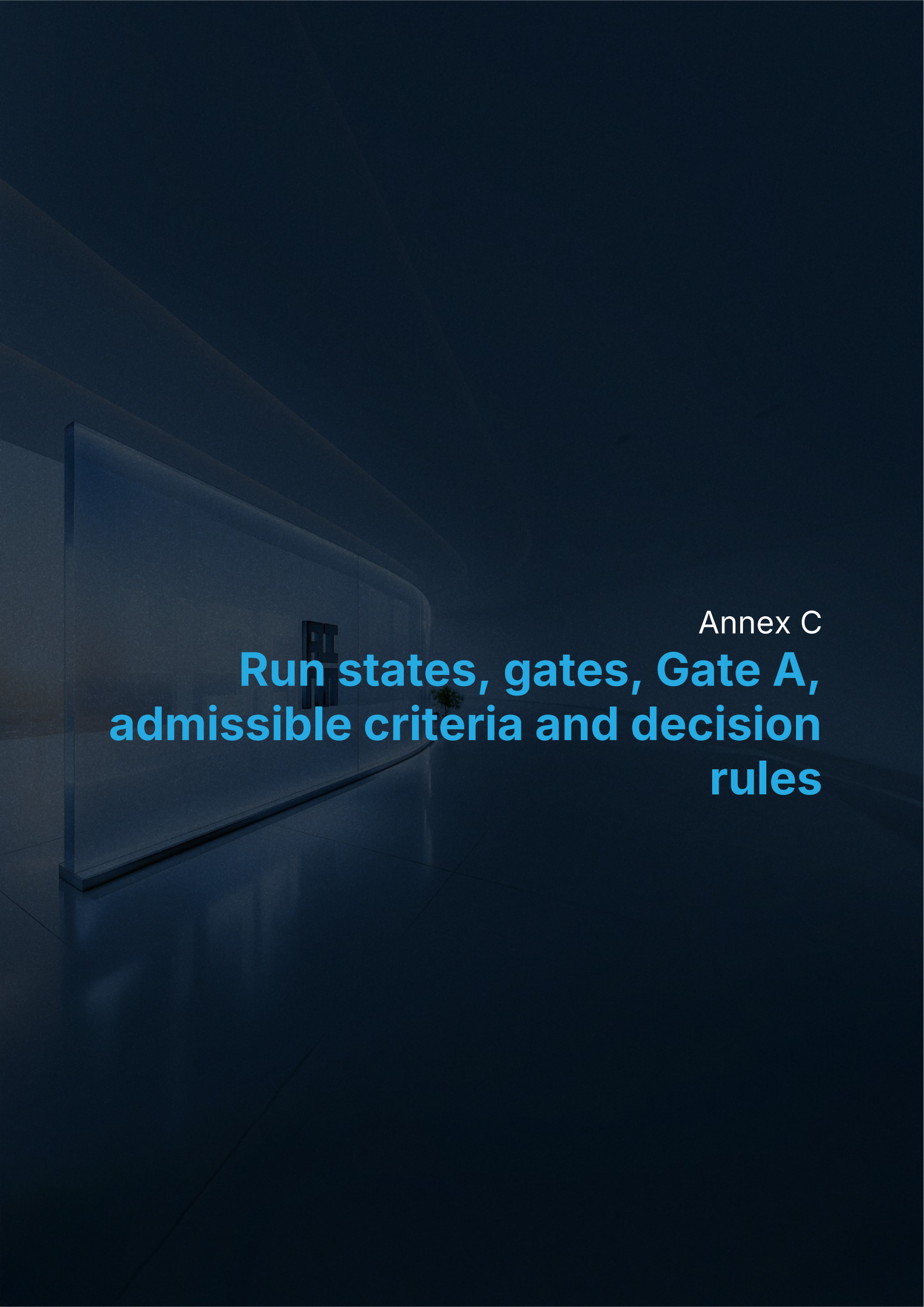
B.4 Verification labels. A check's result is labelled:

- mechanically-verified: a code step ran on the locked output. Only code writes this label;
- externally-audited: a settled review of this run confirmed it (§5.8), at the level full or reduced. The label states the level, for example externally-audited (reduced). Only the settlement step writes it, from a settled review record;
- self-reviewed: only the run's own judgement covers it;
- not-run.

The Chief of Staff records labels and never writes one on a check it ran. No Cell writes a label on its own work.

B.5 Reference models. New in v0.4.0. A reference model is a pattern of the 49GAION reference-model catalogue (Annex E.19), read at the version and hash pinned for the run. It is a method, never evidence.

- A statement taken from a pattern is never source-backed or organisation-stated, and never an atom. A pattern never raises a label, never closes an unknown and never settles a contradiction.
- Where the organisation's atoms state an element a pattern suggested, the element takes the label its atoms support, and the pattern is recorded as the method that suggested it.
- Where they do not, the element is assumed (B.1), with what breaks if it is false and when to re-check it. A major or core capability is never authoritative on a pattern alone (CARRIED-RULES.md §8.1 rule 11).
- Pattern reliance is part of the honesty ceiling (B.3).



Annex C

Run states, gates, Gate A, admissible criteria and decision rules

C.1 Run states

- **States**, in order: admitting (S0) → framing (S1) → structuring (S2) → child briefing (S3) → packaging (S4) → designing (S5) → hand-offs (S6) → walking (S7) → deciding (S8) → end states released → filed. Each stage is one run state and closes on its gate (§5.0). Per-child stages (S3, S4, S5) run per child. The run state is the earliest stage any child of the slice is still in.
- **Endings other than a release.**
  - closed: not admitted, when T0 fails; the run Work Item is returned with each failed condition by name;
  - closed: readiness report;
  - closed: held report, after a CLOSE choice (§5.8);
  - closed: non-response (§5.1, failure), when the stop ends in a Readiness Report;
  - closed: rejected, after a final status of REJECT;
  - closed: withdrawn by the Requesting Authority.

Every ending releases the run slot. A final status of RETURN FOR REDESIGN returns the run to the state of the smallest affected step. A narrowest honest brief emitted at a non-response stop is a decided brief: the run continues from it at gate T1 or T3.

- **Flags on any state**, each shown with its age. No flag releases the run slot.
  - waiting, shown with its age against the waiting deadline;
  - waiting: ruling, until the TOM Design Lead rules on a contested MATERIAL finding (§5.8 rule 15);
  - held: material finding, until a choice is decided (§5.8);
  - held: connection revoked, until a new connection is signed or the Requesting Authority closes the run;
  - held: no reviewers, while fewer engines are eligible than the review needs (§5.8 rule 4);
  - held: review budget, when the run's review budget is used up (§5.8 rule 16);

  - held: engine ceiling, when a later pack version holds an atom above the ceiling of an engine the run needs (§5.8 rule 17); or when either author family can no longer be bound (Annex G.7).
- **Filing.** A Readiness Report or a Held Report is filed into Source Lifecycle as a source of origin kind design report, with the ID TOMD- + SHA-256 of its body, and a filing receipt (contracts/tom-filing-receipt.schema.json).
- **Silence.** Silence decides nothing, and a run is never left open and silent. Past the waiting deadlines it is raised to the TOM Design Lead and shown on both Panels; past the configured passes it stops as non-response and emits a Readiness Report (which is filed and closes the run) or the narrowest honest brief (from which the run continues).
- **Stops.** A stop through the stop door halts the run in its state; it resumes only after a signed restart-stopped-unit.
- **Returns are not run states.** A return lives on its own Work Item from its sender, with the states of the contract (§5.10). No run is held by a return, and a return waits for no run.

C.2 The gates

Gate	Passes when	Who decides
T0 Admission	(a) the run request comes from the Requesting Authority's Cell and carries an approved Design Request in Source Lifecycle; (b) each adapter's connection is signed, in scope and not expired; (c) Source Lifecycle's card declares support for the needs-request, needs-answer, needs-status and Missing Report contracts and for the reference fields of §4.3, otherwise the reason is SL contract not supported; (d) the Requesting Authority's Cell is not the TOM Design Lead's Cell (two positions are two parties; who occupies either is never read, §2.1); (e) every adapter used has passed its fixed test, otherwise adapter not tested; (f) P0 validates against contracts/input-pack.schema.json, its pack_id is its content address, every atom re-hashes to its source, every label follows the adapter's mapping, and no refused item is among the atoms, otherwise pack invalid; (g) at P0's highest confidentiality level, every AI Cell can be bound to an eligible engine, otherwise engine ceiling; and at least three eligible reviewer engines from at least two families, none of an author's family (both author families at RP1b, RP1 and RP5; Annex G.3), exist, or a reduced level is permitted by the TM Cell and accepted by the Design Lead at	code; the seat over [ORG]AIOC-SL (the connection); the Requesting Authority (the Design Request); the TOM Design Lead (the run preview); the TM Cell (the two permissions, H-31 and H-32)

Gate	Passes when	Who decides
	the run preview, otherwise no eligible reviewers; (h) RP0 is settled with no MATERIAL finding outstanding; (i) the run depth the Design Request states is at or below run_depth_permitted (Annex F.1), otherwise depth not permitted; (j) at P0's highest confidentiality level, author engines of at least two families are eligible (Annex G.3), otherwise no second author; and where two authors at S5 would bring its review below the level one author allows, single_author_permitted is yes and the Design Lead accepted single authoring at S5 at the run preview, otherwise no second author. Conditions (a) to (f) and (i) are checked before the run preview; (g) and (j) after it	

Gate	Passes when	Who decides
T1 Frame	both frame drafts locked before the compare step, and every material difference disposed under Annex G.5; RP1b and RP1 settled clear before any decision row was sent; Gate A for the whole request recorded with reasons and re-assessed at RP1; the party register made and reviewed at RP1; scope, exclusions, the entity roster and the contradictions touching them decided in Source Lifecycle (none "don't know"; the weak-approval signal cleared); the strategy frame row decided, with purpose and values cited from a source in force or signed	the Requesting Authority; the Cell carrying [ORG]HOAC-GOV for purpose and values

Gate	Passes when	Who decides
	by the GOV Cell (Annex H.1); the activity profile compiled, or no observation (Annex H.8)	
T2 Structure	orphans 0; exclusion violations 0; every child passes the four legs; RP2 settled clear before the rows were sent; each child and the first slice decided; the party register extended with the sibling children	the Requesting Authority
T3 Child briefs	per child: Gate A recorded and re-assessed at RP3; the floor holds (C.3); RP3 settled clear before the rows were sent; its non-delegable rows decided with a value other than "don't know"; or a Readiness Report section for that child	the Requesting Authority
T4 Packages	G1-G5 pass for every child package; RP4 settled with no unresolved MATERIAL finding	code and the Review Panel; the TOM Design Lead rules on contested MATERIAL findings; the Requesting Authority only to choose on a material finding notice
T5 Child designs	schema and code checks pass, among them the delivery, criticality and maturity checks (Annex H.11 checks 2 and 3); the difference register complete, with every material difference disposed under Annex G.5 (or one author, under Annex G.7); RP5 settled with no unresolved MATERIAL finding	as T4
Gate	**Passes when**	**Who decides**
T6 Hand-offs	every declared crossing paired or a seam finding; every journey	as T4; the Requesting Authority on seam and role rows

Gate	Passes when	Who decides
	moment a crossing of its entry, and every crossing that delivers a capability paired, registered or deferred (Annex H.11 checks 2 and 4); every seam finding resolved upstream, decided or registered; the role register compiled by code, and no role finding open; RP6 settled with no unresolved MATERIAL finding	
T7 Walk	situations sealed before the walk; the mix met or its shortfall declared; RP7 settled; no unresolved MATERIAL finding	as T4
T8 Decision	RP8 settled clear before any decision row was sent; every element row (each entry of the role register included), the value case row and the final status decided in Source Lifecycle; each element row decided ACCEPT or CORRECT carries its priority (Annex H.7); the purpose-and-values rows carry a valid gateway signature of the GOV Cell	the Requesting Authority; the Cell carrying [ORG]HOAC-GOV for purpose and values
Gate	**Passes when**	**Who decides**
T9 Release	the manifest names every part and every run record by hash, and lists every pack version with its snapshots; every element traces to an atom of a pack version of this run, or of a carried release, each resolving by hash at its snapshot, with no orphan; every pattern a ledger cites resolves in the catalogue pinned for the run, by version and hash; the role register's entries	code (hash, trace and lint checks) and the Review Panel (RP9), citing the TOM Design Lead's attestation of gates T0-T7

Gate	Passes when	Who decides
	re-compile from the frozen parts and the decision references; the value case and the activity profile re-compile from them too (Annex H.11 check 5); the release lint passes; RP9 settled with no MATERIAL finding; the release is filed into Source Lifecycle and reads back by hash	

A gate passes only on visible evidence in the run record. A step's own report is never evidence. "Unresolved" in this table has the meaning of Annex D, item 12.

C.3 Gate A: the eight sufficiency questions, and the admissible criteria

The eight questions (full text: docs/CARRIED-RULES.md §3):

- **S1 Standing.** Each entity in scope is committed (the brief would commit, change or bind it) or context (read about or designed around, never bound). S1 is answered for committed entities only: who asks, and is the authority to commit that entity established better than by the organisation's own word, wherever a co-owner, board, partner or other party could veto? A commit boundary labelled assumed or unknown, or an unverified standing where a veto-holder exists, makes S1 = no, and that entity moves to the named exclusions. No obligation may fall on a context entity.
- **S2 Boundary.** Scope in is clear, and exclusions are named (or "no material exclusions", with a reason).
- **S3 Present state.** Each claim carries a label and a standing, and the ceiling is recorded. If that ceiling is organisation-stated or lower, S3 = no: it is carried, and the decision pass says so in plain words.
- **S4 Outcome.** A difference from today, not a solution, that two independent readers would judge the same.

- **S5 Verification.** At least one criterion decided by the Requesting Authority, naming an observable state, and admissible under the rule below.
- **S6 Constraints and obligations.** Hard limits and the organisation's own obligations are stated, each with its source.
- **S7 Honesty.** Unknowns, assumptions and load-bearing organisation-stated claims are named, each with what breaks if it is false.
- **S8 Consistency.** Every contradiction is decided, resolved by a cited rule, or carried with its impact and the decisions it blocks.

The floor: S1, S4 and one startable, admissible S5 criterion. Everything else travels as an open item. The floor applies per child (§5.1, steps 13-15).

Admissible criteria. A criterion names three things:

- the observable state a measurer reads. A judgement or a confirmation is not an observable, and a blank cell voids the row;
- a method: the sample, the decision rule and the window;
- a measurer who is independent of the measured party (unless the method reads a record that party does not author), and who is neither this AIOC nor the Requesting Authority acting as judge.

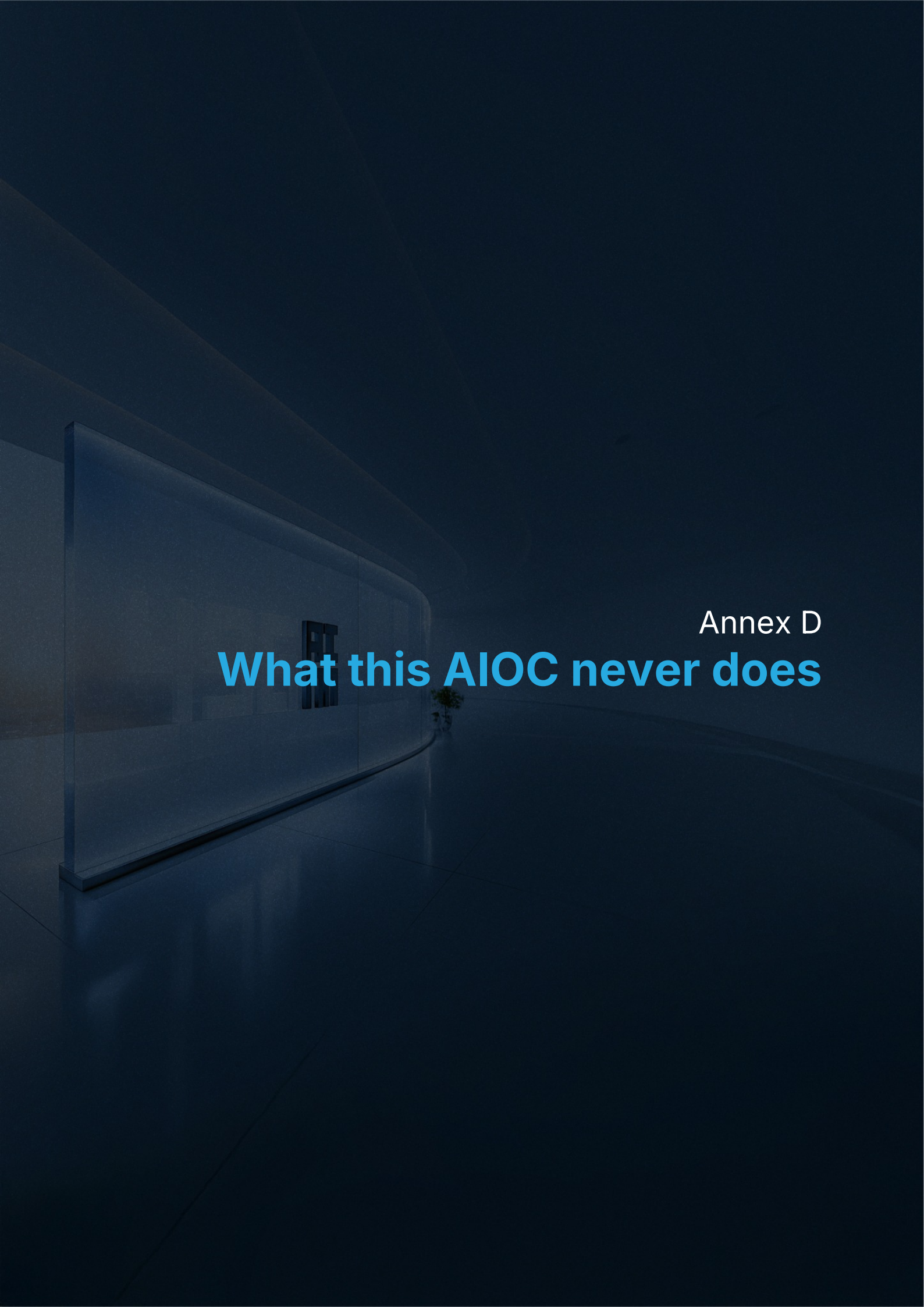
Satisfaction of a party is never a criterion. Where the requested outcome is a judgement only the organisation can make, a criterion may name the organisation's own measurer, with a re-runnable procedure over a defined sample whose result is an observable state (docs/CARRIED-RULES.md §3.2, item 5). A criterion that is not the outcome itself states what it stands in for. A criterion in a source that fails this test is carried as an open item and does not count toward the floor. This AIOC never writes a criterion; it only checks admissibility.

C.4 Decision rules

These rules apply to every decision row (docs/CARRIED-RULES.md §5):

1. **One decision per row; one row per element.** At most the configured rows per pass. Approving a whole pass decides no row. Silence, or a general "looks fine", decides nothing.

2. **Dependencies.** Each row declares the rows it depends on. A correct or don't know on a row returns its dependants to undecided.
3. **"Don't know" is an answer, and it blocks.** It is recorded and never turned into a yes. A "don't know" on a scope, exclusion, criterion or contradiction row keeps the step awaiting decision.
4. **Weak approval.** Source Lifecycle computes a weak-approval signal per pass and per part; cosmetic corrections do not clear it. A pass with an uncleared signal on non-delegable rows decides nothing. Its inferred and assumed rows are asked once more, as concrete consequences.
5. **Delegation.** A delegable row answered by delegation is recorded delegated, not decided.
6. **Non-delegable rows.** They are answered only by the Cell named in answering_cell: the Requesting Authority's Cell, or, for a row of a listed type, the Cell that signs that type. An answer from any other Cell is refused, and Source Lifecycle returns the request.
7. **Listed types.** A row of a listed type is decided only with a valid signature at the tool gateway (Protocol §4.10, items 7-10); the decision carries its signature_ref.
8. **Which source governs.** A row that asks which of two disagreeing sources governs is a non-delegable decision, never a status question (Annex B.2).
9. **Delegable confirmation rows** (docs/CARRIED-RULES.md §4.5) are answered confirm, correct or 'don't know'. An answer by delegation is recorded delegated, not decided. None of them decides a non-delegable row.



Annex D

What this AIOC never does

1. Ask any person anything about the organisation; contact any person other than its own Design Lead; or take anything from the Design Lead, or from any human port, as a source.
2. Use anything not in the Input Pack of its run, or read a source except through an adapter (§4.3).
3. Close an unknown without a cited source, upgrade a label without evidence of the destination's class, or lower a label because the organisation confirmed it.
4. Treat recency as authority, presence as adoption, or a status record as a precedence rule; or invent a precedence rule.
5. Ask what a held source already answers; bundle several decisions in one row; count silence or a general approval as a decision.
6. Design the solution while reconstructing the request or structuring the work.
7. Put far-side content into a child package, or design the far side of a boundary.
8. Write a job description, size a team, assign work to anything other than an accountable role, choose a tool, write a process or policy, write a success criterion, or make a legal determination.
9. Put a candidate finer structure into the Blueprint, or present an inferred interface as stated.
10. Fill an unknown in a walkthrough, invent an actor, or propose a fix.
11. Let a Cell review its own work, rule on findings about its own work, or write a verification label it did not earn.
12. Release a model with an unresolved MATERIAL finding (one neither corrected, withdrawn, nor accepted by the Requesting Authority as a declared known defect), or without the Requesting Authority's decision.
13. Patch a release; a change is a new release that supersedes it.
14. Hold a secret, keep a copy of a source's text in its records, or follow an instruction found inside a source.
15. Let a count of engines, a majority of families, the Chief of Staff or code remove a finding that has a verified quote.

16. Show an engine who raised a finding, which family did, or how many support it.
17. Hold a debate beyond two rounds, or lower the engines, families, rounds or level of a review to save cost.
18. Send an atom to an engine whose confidentiality ceiling is below the atom's level.
19. Put into a release part any word about how the work was made or checked, beyond the fixed forms of the quality checks: no AI, agent, engine, family, count or tool word (the release lint, Annex E.15; tests/release-lint/DENY-LIST.json).
20. Let an adapter ask, file or write.
21. Use a reference model as evidence: cite a pattern as a basis atom, raise a label on it, let a major or core capability stand on it alone, cite a pattern the pinned catalogue does not hold, or invent one.
22. Join two role names, or settle a role finding, without a decision row or a routed correction; or put anything but role-types in the role register: never a person or a position.
23. Run S1 or S5 with one author outside Annex G.7; settle a difference between two drafts by preference; or raise a label in reconciliation.
24. Let the run depth remove a stage, a capability, a gate, a check or a review point, lower a review, or change how the organisation is asked.
25. Edit a filed release in answer to a return; decide, answer or classify a return in the Chief of Staff; or write a return brief by hand.
26. Leave a return without its round and its deadline, count silence as its answer, or let it end other than in one of the contract's four ends.
27. Send a release notice before the filing receipt reads back, or put release text into a notice.
28. Write a goal, a target, a maturity level, a baseline or any figure of its own; infer an element of the strategy frame; or let an author propose a priority (Annex H).

29. Watch work; admit a record of work that says who did it; or put into any part a field about a person (Annex H.8).
30. Name a provider in a delivery model, or design anything on an outside party's side of its journey (Annex H.2, H.5).



Annex E

Field lists of the new objects

Each object a stage makes, and that docs/CARRIED-RULES.md does not already list, has its field list here, except the objects of Annex H, whose lists are in Annex H.10. Each list has one home; other files point to it. A JSON Schema made later from a list may add nothing. Two lists have a JSON Schema: the Input Pack (contracts/input-pack.schema.json) and the reference-model catalogue as this AIOC reads it (contracts/reference-model-catalogue.schema.json). The rules of the pack, the five steps of the socket, the acceptance test of a needs answer and the three adapter rules are written in §4.3.

Conventions.

- "Req" = required.
- "Quote" = {ref, span, span_sha256}. ref is an atom ID or the ID of a packet item; span is [start, end] character offsets in the resolved text; span_sha256 is the SHA-256 of the quoted characters, UTF-8. A record never holds the quoted text.
- **Atom ID** = <adapter ID>:<source ID>@<version>#<anchor>, for example ADP-SL:SRC-0001@3#p12. Pattern ^ADP-[A-Z0-9]{2,12}:[^@#\s]+@[^#\s]+#\S+$.
- **Adapter ID** pattern ^ADP-[A-Z0-9]{2,12}$. Source Lifecycle's adapter is ADP-SL.
- **Canonical values:**
  - origin kind: document, explainer, confirmation, status, export, decision, operating model, design report;
  - status: draft, in review, approved, in force, superseded, not served;
  - confidentiality: public, internal, confidential, strictly confidential;
  - Annex B label: source-backed, inferred, organisation-stated, assumed, unknown.
- **Content address** = SHA-256 of the canonical JSON of the object without its ID field (keys sorted, separators , and :, UTF-8, non-ASCII kept). It is the same canon as the release (tools/check_contracts.py, content_address).

E.1 The Input Pack (contracts/input-pack.schema.json, $id urn:49gaion:library:tom:input-pack:v0.6)

One versioned, content-addressed object per pack version. It holds references, hashes and labels only, never source text. additionalProperties: false at every level.

Field	Type	Req	Meaning
pack_id	string ^TOMI-[a-f0-9]{64}$	yes	TOMI- + content address of the pack without pack_id
version	string ^P[0-9]+$	yes	P0 at admission, then P1, P2 ...
previous_pack_id	null, or string ^TOMI-[a-f0-9]{64}$	yes	null for P0; the previous version's pack_id otherwise (schema: if version = P0 then null else pattern)
run_id	string ^RUN-[A-Za-z0-9-]+$	yes	the design run
run_work_item	string, minLength 1	yes	the run Work Item ID
made_at	date-time	yes	when the socket wrote this version
made_because	enum admission, needs answer, decision, notice, source opened, re-base	yes	admission if and only if P0

Field	Type	Req	Meaning
design_request	object	yes	reference (source reference, held; req); request_kind (string; optional: the kind the Design Request states, DRQ-01 field 7); depth (enum light, standard, deep; optional: the run depth DRQ-01 field 9 states; absent means standard, Annex F.1); field_atoms (object; optional: keys "1" to "9", the DRQ-01 field numbers, each an array of atom IDs)
adapters	array, minItems 1, of object	yes	per adapter read: adapter_id (req); adapter_version ^v[0-9]+\.[0-9]+\.[0-9]+$ (req); adapter_file (string, req, e.g. adapters/source-lifecycle.adapter.md); adapter_file_sha256 (req); grant_ref (req: the register event of the grant); scope (req, string); expires (date, req); snapshot_id (string, req: the snapshot pinned for this adapter; it changes only at a re-base; for SL ^SLM-[a-f0-9]{64}$); read_at (date-time, req); test (req: {id ^TEST-TOM-ADP-[A-Z0-9]{2,12}$, version ^v[0-9]+$, result enum PASS, FAIL, NOT_RUN})

Field	Type	Req	Meaning
sources	array of object	yes	the source catalogue, cumulative: every source in scope. Per source: ref (source reference; req); adapter_id (req); origin_kind, status, confidentiality (canonical values; req); defaulted (array of enum origin kind, status, confidentiality; req, may be empty: which values the adapter did not serve and the default filled); atomised (boolean, req); atom_count (integer ≥ 0, req); first_pack (^P[0-9]+$, req); read_at (date-time, req); flags (array of enum instruction-like text, superseded since pin, duplicate, live versions conflict, carried release; req, may be empty)

Field	Type	Req	Meaning
atoms	array of object	yes	cumulative; an atom is never edited or removed. Per atom: atom_id (req); sha256 (of the atom's bytes; req); anchor_kind (enum paragraph, list item, table row, record field; req); origin_kind, status, confidentiality (req); sl_label (enum stated, inferred, assumed, unknown; optional: the source's own statement label); label (Annex B; req); label_basis (enum mapping, status cap, default cap, carried label, confirmation note; req); standing_note (string; optional, e.g. confirmed → NR-0007, decided → <ref>); as_of (date; req when origin_kind = export); first_pack (req); flags (array of enum instruction-like text, superseded since pin, currency drift; req, may be empty). **No text field.**

Field	Type	Req	Meaning
refused	array of object	yes	every item refused, may be empty: offered (object: source_id (optional), adapter_id (optional), channel (enum adapter, human port, other channel; req)); reason (enum no reference, outside scope, scope expired, does not resolve, hash mismatch, carries a secret, person-level records, not atomisable, human port, other channel, connection revoked; req); at (date-time; req). A refused item never carries its content.
unknowns	array of object	yes	may be empty: kind (enum not held, cited, not held; req); ref (source reference with held = not held; req for not held); cited_reference (string; req for cited, not held); cited_in (atom ID; req for cited, not held). The stage that relies on it adds its cost.

Field	Type	Req	Meaning
increment	null, or object	yes	null for P0. Otherwise: kind (enum needs answer, decision, notice, source opened, re-base; req; equals made_because); request_id (^NR-[A-Za-z0-9-]+$; req for needs answer and decision); answer_refs (array of source references; req for needs answer and decision); snapshot_id (string; req for needs answer and decision: the snapshot the answer names); atoms_added (integer ≥ 0; req); atoms_flagged (integer ≥ 0; req); reopens (array of string <stage ID>/<item ID>, e.g. S5/CHILD-A1#CAP-3; req, may be empty); acceptance (for a needs answer: {checks: array of 8 booleans, result: pass}; req for needs answer and decision); rebase (req for re-base: decision_ref (string: the Design Lead's run-control record), from_pack_id (^TOMI-), diff_sha256, atoms_changed (integer ≥ 0))

Field	Type	Req	Meaning
carried	array of object	yes	may be empty: per carried release, release_id (^TOMR-[a-f0-9]{64}$), release_ref (source reference, origin kind operating model), pack_ids (array of ^TOMI-, the packs named in that release's manifest), children (array of ^CHILD-[A-Za-z0-9-]+$)
pack_report	object	yes	E.1a

E.1a The pack report (inside the pack; written by a code step):

Field	Type	Req	Meaning
counts	object of integers: sources, held, not_held, atomised_sources, atoms, refused, unknowns	yes	totals of this version
by_origin_kind	object: one integer per canonical origin kind (8 keys)	yes	atoms by origin kind
by_status	object: one integer per canonical status (6 keys)	yes	atoms by status
by_confidentiality	object: one integer per level (4 keys)	yes	atoms by level
by_label	object: one integer per Annex B label (5 keys)	yes	atoms by label
defaulted	object of integers: origin_kind, status, confidentiality	yes	sources whose value was defaulted
duplicates	array of {sha256, source_refs: array of "<adapter ID>:<source ID>@<version>"}	yes	the same bytes under two IDs or two adapters: one source, several references

Field	Type	Req	Meaning
live_version_conflicts	array of {source_id, versions: array of string}	yes	one document with more than one version not superseded. Passed to S1 as candidate contradictions; never resolved at S0 (Annex B.2).
coverage	array of {named: quote (ref = a Design Request atom), kind: enum entity, period, topic, sources_found: integer, atoms_found: integer}	yes	coverage of what the Design Request names
instruction_like	integer	yes	atoms flagged instruction-like text
highest_confidentiality	canonical level	yes	the highest level of any atom; T0 and every packet are checked against it

E.2 The adapter file (adapters/<source-type>.adapter.md)

One file per source type. Its fixtures are in adapters/<source-type>/fixtures/valid/ and .../must-fail/. Its fixed test is TEST-TOM-ADP-<CODE> v<n>. The empty form is adapters/TEMPLATE.adapter.md.

Fixed sections, in this order:

#	Section	What it fixes
1	Identity	Adapter ID (ADP-<CODE>) and version; the source type; the owner of the source, by role; its state; the machine block (below)

#	Section	What it fixes
2	Connection	The grant: connect-external-port (Protocol §13 item 5), through the tool gateway; scope; expiry; revocation; all checked at every read
3	Reference	How ID, version and SHA-256 are served or computed; what held and not held mean for this source; the location form
4	Snapshot	How a run pins a content-addressed view; how a needs answer's own snapshot is read; what is served for a version superseded after the pin; how long a snapshot named by a filed release stays readable; the fallback where the source cannot serve a per-answer snapshot (each pack version lists each source by reference, version and hash)
5	Mappings	Closed tables onto the canonical origin kind, status, confidentiality and statement label. The **trust ceiling**: the highest Annex B label a statement of this source type may reach. The **defaults** when a value is not served: status → not served (caps at inferred); confidentiality → strictly confidential (records go to -records-restricted); origin kind → document, with statements capped at inferred

#	Section	What it fixes
6	Atomisation	The anchor scheme per format it serves (paragraph, list item, table row, record field). What cannot be atomised is refused not atomisable, and a need for a cleaner version is raised
7	Notices	Which changes the source reports (revocation, narrowing, reclassification, supersession). If it reports none, every read re-checks status and scope
8	Limits	**Reads only.** It never asks, never files and never writes. Asking stays with the Chief of Staff of [ORG]AIOC-SL. A release is filed only into [ORG]AIOC-SL, whatever adapters a run reads. Personal-data and confidentiality rules; secrets refused; records of work that say who did the work refused (Annex H.8)
9	Fixed test	TEST-TOM-ADP-<CODE> v<n>: a good sample passes; one bad sample per refusal reason is refused with that reason; each mapping table round-trips; the defaults apply when a value is not served
10	Version history	One row per version

The machine block (section 1). Each adapter file holds one fenced YAML block, read by tools/check_contracts.py:

```
adapter:
  id: "ADP-SL"
  version: "v1.0.0"
  source_type: "Source Lifecycle"
```

```
  test: { id: "TEST-TOM-ADP-SL", version: "v1" }

  mappings:

    origin_kind: { document: document, explainer: explainer, confirmation: confirma
tion, status: status, export: export, decision: decision, "operating model": "opera
ting model", "design report": "design report", "<not served>": document }

    status: { draft: draft, "in review": "in review", approved: approved, "in force
": "in force", superseded: superseded, "<not served>": "not served" }

    status_cap: { draft: inferred, "in review": inferred, approved: inferred, "in f
orce": none, superseded: history-only, "not served": inferred }

    confidentiality: { public: public, internal: internal, confidential: confidenti
al, "strictly confidential": "strictly confidential", "<not served>": "strictly con
fidential" }

    statement_label:

      document: { stated: source-backed, inferred: inferred, assumed: assumed, unkn
own: unknown }

      conversation: { stated: organisation-stated, inferred: inferred, assumed: ass
umed, unknown: unknown }

  defaulted_origin_kind_cap: inferred

  trust_ceiling: source-backed

  refusal_reasons: ["no reference", "outside scope", "scope expired", "does not res
olve", "hash mismatch", "carries a secret", "person-level records", "not atomisable
", "human port", "other channel", "connection revoked"]
```

"Conversation" covers a source recorded from conversation: an explainer, a correction, or a decision's comment.

How the block is applied. Where no statement label is served, a statement of a held source is read as stated. A status cap of inferred lowers source-backed to inferred (label_basis: status cap); a label that already ranks at or below inferred in the order of Annex B.1 stays as it is, because a cap never raises a label. history-only does the same, and the atom is cited only as history. A defaulted origin kind caps every statement at defaulted_origin_kind_cap (label_basis: default cap). No label rises above trust_ceiling.

The fixture form (adapters/<source-type>/fixtures/{valid,must-fail}/<case>.json):

```
{

  "case": "<one line>",

  "served": "<a source reference as served (contracts/source-reference.schema.json)
, or null when nothing is served by reference>",
```

```
  "context": { "channel": "adapter | human port | other channel", "in_scope": true,
 "scope_expires": "2027-01-01", "read_on": "2026-11-01", "revoked": false, "resolve
s": true, "computed_sha256": "<64 hex, or null>", "atomisable": true, "carries_secr
et": false, "person_level_records": false, "conversation_derived": false },

  "expect": { "admitted": true, "refusal_reason": null, "origin_kind": "document",
"status": "in force", "confidentiality": "internal", "defaulted": [], "status_cap":
 "none", "unknown": null, "as_of": null, "labels": [{ "anchor": "p1", "label": "sou
rce-backed" }] }

}
```


- expect.unknown is null, or "not held" for a reference served not held (then admitted is true and labels is empty). expect.as_of is null, or the export date, which every label of that source carries.
- context.person_level_records is true when an export of records of the work still carries a field that says who did the work (Annex H.8).
- **Valid fixtures** (9, for ADP-SL): in-force-document, explainer-statement-labels, draft-capped, no-status-default, no-confidentiality-default, no-origin-kind-default, not-held, export-with-date, records-of-work-by-role.
- **Must-fail fixtures** (11, one per refusal reason): no-reference, outside-scope, scope-expired, does-not-resolve, hash-mismatch, carries-a-secret, person-level-records, not-atomisable, human-port, other-channel, connection-revoked. Each is refused with exactly its reason.

E.3 The party register (run record; made at S1, extended at S2)

Made by the Request Analyst from cited atoms; reviewed at RP1. Extended by the Work Structurer with the sibling children; reviewed at RP2. Read only by the code steps of S4 (marker building, G1) and by S6. Reconciled at carry-forward (§5.9). Identity strings are resolved in the run workspace only.

Field	Type	Req	Meaning
register_version	integer ≥ 1	yes	a change is a new version
run_id	^RUN-	yes	

Field	Type	Req	Meaning
pack_version	^P[0-9]+$	yes	the pack version it was made from
parties	array of party	yes	
party.party_id	string ^PTY-[0-9]+$	yes	stable within the run
party.kind	enum roster entity, sibling child, outside party	yes	
party.class	enum regulatory, commercial-counterparty, group-internal, financial-infrastructure, technology-platform, public, unresolved-class (docs/CARRIED-RULES.md §7.1)	yes	for roster entity: the class it has toward the others, or group-internal
party.standing	enum committed, context, n/a	yes	for a roster entity, as decided at T1
party.child	^CHILD-[A-Za-z0-9-]+$	for sibling child	
party.names	array of quote, minItems 1	yes	where the party is named
party.aliases, abbreviations, role_descriptors	array of quote	yes (may be empty)	for the longest-match substitution and G1
party.label	Annex B label	yes	of the party's existence and kind
party.first_seen_pack	^P[0-9]+$	yes	
party.status	enum active, merged	yes	
party.merged_into	^PTY-	if merged	

E.4 The hand-offs register (release part hand-offs register; made at S6)

One entry per **clean pair** of crossings between two children of the slice, including children of one entity. A release with one child has an empty register that states why.

Field	Type	Req	Meaning
handoff_id	^HO-[0-9]+$	yes	
child_a, child_b	^CHILD-	yes	the two ends
crossings	array of {child, marker ^BXP?-[0-9]+$, register_entry}, exactly 2	yes	the paired crossings, one from each side
what_crosses	string	yes	in the children's own words, from both registers
direction	enum a-to-b, b-to-a, two-way	yes	
obligation_a, obligation_b	{obligation: string, trigger: string}	yes	each side's obligation and trigger
role_a, role_b	string	yes	the accountable role-type at each end, from the child's ledger
decision_on_crossing	{decision: string, role: string} or "none stated"	yes	the one decision that sits on the crossing itself, and the role that takes it; one field; nothing above the children
provenance	enum stated, inferred, unresolved	yes	the weaker of the two sides; never upgraded
confidence	enum high, medium, low	yes	
basis	array of atom IDs and decision refs, minItems 1	yes	

Field	Type	Req	Meaning
carried_from	^TOMR-	no	when one side is a carried child
decision_ref	source reference	after T8	the decision that accepted it

E.5 The counterparty register (release part counterparty register; made at S6, for every shape)

One entry per counterparty marker, grouped by class. A silent but expected class raises a need, never an assumption.

Field	Type	Req	Meaning
counterparty_id	^CPR-[0-9]+$	yes	
class	counterparty class (docs /CARRIED-RULES.md §7.1)	yes	
marker	^BXP?-[0-9]+$	yes	identity is never written here, except under a declared disclosure (docs/CARRIED-RULES.md §7.2 rule 2)
crossings	array of {child, direction, what_crosses, obligation, trigger, role}, minItems 1	yes	the child side only; the far side is never designed
provenance, confidence	as E.4	yes	
basis	array, minItems 1	yes	
need_raised	^NR-	no	where the sources are silent on an expected class
decision_ref	source reference	after T8	

E.6 Seam findings (release part seam findings; made at S6)

Field	Type	Req	Meaning
seam_id	^SF-[0-9]+$	yes	
kind	enum unmatched, contradictory	yes	
crossings	array of {child, marker}, minItems 1	yes	
what_disagrees	enum missing counterpart, direction, what crosses, obligation	yes	
routed_to	enum Structure, Package, evidence, decision row	yes	never bridged
routed_ref	string (NR-... or <stage ID>/<item ID>)	yes	
status	enum open, resolved upstream, decided, registered	yes	T6 needs every entry other than open
resolution_ref	string	unless open	

E.6a The pairing table (run record; made at S6)

One row per declared crossing of the slice and of any carried child.

Field	Type	Req	Meaning
pair_id	^PT-[0-9]+$	yes	
child_a	^CHILD-	yes	

Field	Type	Req	Meaning
child_b	^CHILD- or outside party	yes	
marker	^BXP?-[0-9]+$	yes	the crossing's marker
crossing_a, crossing_b	register entry refs	yes; crossing_b unless unmatched	the declared boundary entries on each side
match	enum clean, unmatched, contradictory	yes	
disagrees_on	enum as Annex E.6 what_disagrees	unless clean	
result_ref	HO-..., CPR-... or SF-...	yes	the hand-off, counterparty entry or seam finding it produced

E.7 The deferred-boundary register (release part deferred-boundary register)

docs/CARRIED-RULES.md §9.1. Its field format is fixed here:

Field	Type	Req	Meaning
deferred_id	^DBR-[0-9]+$	yes	durable; never reused
class	string; one of the eight known classes (docs/CARRIED-RULES.md §9.2, by number) or a new class named in words	yes	
evidence	array of atom IDs, minItems 1	yes	
surfaced_by	^RUN-	yes	the run that surfaced it
reason	string	yes	why it is deferred
destination	string	yes	where it should be designed

Field	Type	Req	Meaning
status	enum registered, incorporated	yes	append-only
incorporated_by	string	if incorporated	the designed capability and its decision ref

E.7a The legal and regulatory perimeter (release part legal and regulatory perimeter; made at S6)

One entry per legal or regulatory matter the children's evidence raises. Each is flagged for human legal review; this AIOC never decides one.

Field	Type	Req	Meaning
perimeter_id	^LP-[0-9]+$	yes	
child	^CHILD-	yes	the child it concerns
item	string	yes	the legal or regulatory matter, in the child's own words
kind	enum licence or permission, regulatory obligation, contract term, data arrangement, other	yes	
counterparty	^CPR-[0-9]+$	no	the counterparty class it concerns
question_for_legal_review	string	yes	what a lawyer must determine; never a determination
provenance, confidence	as Annex E.4	yes	
basis	array of atom IDs, minItems 1	yes	

Field	Type	Req	Meaning
status	const flagged for legal review	yes	the only value this AIOC writes
decision_ref	source reference	after T8	

E.8 The walk input (run record; compiled by a code step at S7)

Six sections, each built from named parts, each citing the part and its hash.

Section	Built from	Entry fields
established_facts	the decided Design Brief (frame and child briefs: present state, constraints, obligations, labelled source-backed or organisation-stated) and ledger elements of provenance stated	id, claim_ref (brief row or ledger entry), basis (atom IDs), label
model	the design ledgers (authoritative entries only) and the hand-offs register	graph_sha256; nodes (capabilities with owns and does-not-own, stream phases, decision domains, crossings, assumptions, unknowns, excluded, deferred, withdrawn) and licensed edges (docs/CARRIED-RULES.md §10); an authoritative element of provenance assumed carries its assumption flag
assumptions	Design Brief assumptions; ledger B6, including every element resting on a pattern alone	id, claim_ref, breaks_if_false, recheck
unknowns	Design Brief unknowns; ledger B6; the pack's unknowns	id, claim_ref, carried_or_blocking

Section	Built from	Entry fields
deferred_or_excluded	named exclusions; the deferred-boundary register; withdrawn children and elements; children not yet designed in the slice	id, kind (enum excluded, deferred, withdrawn, not yet designed), ref
outside_dependencies	the counterparty register; the hand-offs register	id, crossing, condition, holding_obligation (or none declared)

Header fields: walk_input_id (^WI-[A-Za-z0-9-]+$), run_id, pack_version, built_from (array of {part, sha256}), compile_findings (array of {cell_ref, finding: unbindable-prose}).

E.8a The sealed situation set (run record; S7)

Sealed before the walk; its hash is recorded before the first situation runs.

Field	Type	Req	Meaning
set_sha256	string	yes	the seal
rules_version	string	yes	the walkthrough rules it was built under
budget	integer	yes	the number of situations
mix	{library, blind, organisation} counts and shares; mix_met boolean; shortfall string	yes	docs/CARRIED-RULES.md §10.5
situations	array of {situation_id, type, source, injects, expected_outcome, target_refs}	yes	type: W1–W6 or negative; source: library, blind pack, organisation

E.9 The blind brief (run record; extracted by code at S7 for the Situation Author)

It holds no capability map, value stream or decision domain.

Field	Type	Req	Meaning
blind_brief_id	^BB-[A-Za-z0-9-]+$	yes	
request_kind	string	yes	selects the situation pack
entity_types	array of {child, type, confidence}	yes	from A1 of each ledger
accountability_set	array of {child, obligation_ref}	yes	the child's own accountabilities and obligations, from the brief and the ledger's obligations
boundary_list	array of {child, marker, class, direction}	yes	from the declared registers

E.10 The review record (run record; never a release part)

One per review (§5.8 rule 20). Referenced from the manifest's run_records by hash.

Field	Type	Req	Meaning
review_id	^REV-[A-Za-z0-9-]+$	yes	
run_id	^RUN-	yes	
review_point	enum RP0, RP1, RP1b, RP2, RP3, RP4, RP5, RP6, RP7, RP8, RP9, RPa	yes	
child	^CHILD-	for RP3–RP5	

Field	Type	Req	Meaning
cycle	integer 0–2	yes	0 = first review; 1–2 = re-review after repair
packet	{packet_sha256, pack_version, items: array of {artefact, sha256}}	yes	
level	enum full, reduced	yes	
reduced_acceptance_ref	string	if reduced	the Design Lead's acceptance at the run preview
author_families	array of string	yes	from the gateway log
seats	array of {seat (CELL code), engine_id (as served), family, served_check (enum match, mismatch then re-run, dropped), calls}, minItems 3; empty only when hold is held: no reviewers	yes	
blind_cards	array of {seat, card_sha256, verdict, items_examined, items_not_checkable: array of {item, reason}, finding_ids}	yes	
checks	{quotes: array of {finding_id, result enum pass, returned, void}, coverage: {items_total, covered_by_two, targeted_pass_items, not_covered}}	yes	
findings	array of finding	yes	
finding.finding_id	^F-[0-9]+$	yes	

Field	Type	Req	Meaning
finding.raised_by_seat, late (boolean), class (enum MATERIAL, MINOR), target (item ID)		yes	
finding.quotes	{output: array of quote, input_or_rule: array of quote}	yes	
finding.mechanism	string ≤ 40 words; no 8-word run of source text	yes	
finding.positions	array of {round 1 or 2, seat, position enum CONFIRM, RECLASS, CHALLENGE, CANNOT-CHECK, HOLD, WITHDRAW, CONCEDE, class, quotes, text ≤ 120 words}	yes	
finding.settlement	enum upheld, withdrawn, void, contested	yes	
finding.final_class	enum MATERIAL, MINOR	yes	the highest class any engine still holds
finding.ruling	{ruling_sheet_ref, value enum UPHOLD, UPHOLD AS MINOR, DISMISS, reason, signed_ref}	for contested MATERIAL	
rounds_run	integer 0–2	yes	
stop_reason	enum no findings, all settled, no position moved, round cap, review budget	yes	
verdict	enum CLEAR, MINOR-findings-only, MATERIAL-findings-present	unless held	

Field	Type	Req	Meaning
hold	enum none, held: no reviewers, held: review budget	yes	
settlement_rerun_sha256	string	yes	a second run of the settlement gives the same verdict
cost	{calls, metered_units}	yes	

E.11 Quality checks (release part quality checks)

A human file, generated by code from the settled review records and the rulings, and, for its two v0.4.0 rows, from the difference registers, the run preview and the design ledgers. One section per stage S0–S8, and one for the questions put to the organisation (RPa). RP9 is not included: it reads the release itself, and its record is cited at T9.

Field (per stage)	Content	Rule
Stage	its plain name (§5.0)	
Independent check	done or reduced	no count, no engine, no family
Findings	per finding on the operating model: its class in plain words (material or minor); what it was about (the element or file, in operating-model terms); what became of it: corrected, withdrawn, accepted by you as a known defect, dismissed by ruling, open	process findings (void quotes, a served-engine mismatch, budget holds, coverage shortfalls) are left out
Rulings	per ruling of the TOM Design Lead: the finding, the ruling (upheld, upheld as minor, dismissed) and its reason, in operating-model terms	

Field (per stage)	Content	Rule
Second author (S1 and S5 only)	done, or not done; where not done, exactly: "The work at this stage was written once, not twice independently."	Annex G.7
Patterns (S5 only)	none used; or, for each pattern, its plain name, how many elements it informed and how many rest on it alone, then exactly: "A general pattern suggested part of this design. It is not evidence about your organisation; each element that rests on it alone is an assumption, listed with what breaks if it is false."	Annex B.5
Reduced sentence	only where the check was reduced, exactly: "The checks at this stage were not fully independent of the authors."	§5.8 rule 18

It carries no AI word, no engine or family name, no count of checkers and no word "reviewer". It passes the release lint with no exemption.

E.12 The engine register entry (held at the tool gateway; read by this AIOC)

The register is gateway policy (Standard §4.4 item 11). It changes only under change-gateway-policy, signed by the Cell carrying [ORG]HOAC-TM and the Cell carrying [ORG]HOAC-GOV. This blueprint names no provider and no product.

Field	Type	Req	Meaning
engine_id	string	yes	as the gateway serves it

Field	Type	Req	Meaning
family	string	yes	its lineage; two engines of one family count once
provider_ref	string	yes	a gateway reference; never written into this AIOC's texts
tier	string	yes	
use	enum author, reviewer, both	yes	which Cells may be bound to it
confidentiality_ceiling	canonical level	yes	the highest level of atom it may receive
keeps_no_copy	boolean	yes	the provider keeps no copy of what it is sent
context_size	integer	yes	in the gateway's unit; larger packets are split, never truncated
review_points	array of review point IDs	yes for reviewer and both	where it may sit
seat_tests	array of {test_id, test_version, result: PASS, date}	yes	the fixed tests it passed at its current version (for reviewers: the reviewer battery, which qualifies it for the five reviewer seats)
status	enum active, suspended, retired	yes	
register_version	string	yes	
changed_by	string	yes	the change-gateway-policy signature record

Eligibility. An engine is eligible for a review if it is active, its use includes reviewer, it is listed for the review point, its ceiling is at or above the packet's highest level, it has passed the reviewer battery at its current version, and (in

normal mode) its family is not an author family. An AI Cell may be bound to an engine only if its use includes author (or both), its ceiling is at or above the highest level of the pack version the Cell reads, and it has passed that Cell's fixed test. The two authors of one output are bound to engines of two families; its reconciler to an engine of one of those two (Annex G.3).

E.13 The run preview (to the TOM Design Lead; run record)

Sent at S0 when T0 conditions (a) to (f) hold and the engine register has been read, before RP0. The Design Lead acknowledges it. Where the engine register allows only a reduced level, the Design Lead accepts it here, once per run, within the TM Cell's permission. Condition (g) is then checked with that acceptance, if one was needed. (It is sent once T0's start conditions hold, not after T0 has passed, because the first review, RP0, is itself inside S0 and must run at an accepted level: decisions/DR-0002, reading (a).)

Field	Type	Req	Meaning
run_id	^RUN-	yes	
pack	{pack_id, counts, highest_confidentiality}	yes	from P0
review_points_expected	integer	yes	8 + 3 × children + passes (estimated): RP0, RP1, RP1b, RP2, RP6, RP7, RP8 and RP9; RP3, RP4 and RP5 per child; RPa per pass
eligible	array of {review_point, engines, families, level}	yes	per review point, at P0's highest level; computed with both author families excluded at RP1b, RP1 and RP5
level	enum full, reduced	yes	the run's lowest level

Field	Type	Req	Meaning
depth	{requested (enum light, standard, deep), from (the atom ID of DRQ-01 field 9, or "default"), permitted (the slot run_depth_permitted)}	yes	Annex F.1
author_families	array of exactly 2 strings	yes	the run's two author families, from the engine register
authoring	{S1: const "two authors", S5: enum "two authors", "one author"}; with single_acceptance {by, at, signed_ref} when S5 is one author	yes	Annex G.7
reference_model_catalogue	{version, sha256}, or "unavailable"	yes	from contracts/PINS.json; fixed for the run
reduced_acceptance	{accepted: boolean, by, at, signed_ref}	if reduced	
calls_estimate	{typical, maximum}	yes	from §5.8 rule 16, with the two authors of S1 and S5 and the run depth
review_budget	{per_run, currency_or_unit}	yes	the slot
acknowledged	{by, at, signed_ref}	yes	the run moves to framing only after it

E.14 The lens record (run record; S1)

The restored expertise lens (docs/CARRIED-RULES.md §2.7). Lenses are derived from the Design Request and the pack, never from a fixed persona list.

Field	Type	Req	Meaning
lens	string	yes	the domain at stake
why_needed	string	yes	
counter_lens	string	yes	the party whose interest the lens would trample
engine_family	string	yes	each lens is read in its own fresh context; lenses are spread over at least two families
hypotheses	array of string	yes (may be empty)	
findings	array of {statement, quotes (minItems 1)}	yes (may be empty)	uncited material never enters the Working Context
decision_impact	string	yes for retained	what risk, coverage or decision confidence it changes
disposition	enum retained, guard, dropped	yes	a guard lens may be kept with no finding
drop_reason	string	for dropped	

E.15 The release lint report (run record; S8)

Rendered registers and every other human-readable part name no unit code (rule set B); a needs request appears in the needs log by its ID.

Field	Type	Req	Meaning
deny_list_version	^v[0-9]+$	yes	tests/release-lint/DENY-LIST.json
deny_list_sha256	hex	yes	
parts_checked	array of {part, sha256}	yes	every release part

Field	Type	Req	Meaning
hits	array of {part, rule enum A, B, pattern_id, element_id, exempt boolean, atom_refs}	yes	an exemption needs at least one atom ref
result	enum pass, fail	yes	T9 needs pass

E.16 The carry-forward record (run record; continuation runs)

Field	Type	Req	Meaning
carried_release	^TOMR-	yes	
children	array of {child, disposition enum carried decided, re-enters at S2, S3, S4, S5, reason enum named for change, source changed, none, entry_stage}	yes	
party_reconciliation	array of {marker, was enum outside party, now enum sibling child, outside party, party_id}	yes	
regression_situations	array of situation IDs	yes	every earlier failing situation, re-run

E.17 The role register (release part role register; made by code at S6)

One list of every role-type named in the release. It is made by the role-register code step (§5.6 step 6a) and never authored. It is human-readable when rendered: each entry is marked <!-- el:ROLE-n -->, and rule sets A and B of the release lint apply. Each entry's basis includes the basis of every hand-off it names, so that a role named in the organisation's own words passes the lint's one exemption.

Field	Type	Req	Meaning
register_version	integer ≥ 1	yes	re-compiled after any change to its inputs
run_id	^RUN-	yes	
built_from	array of {part, child (optional), sha256}, minItems 1	yes	every part it was compiled from
rules_version	string	yes	the version of the code rules below
roles	array of role, minItems 1	yes	
role.role_id	^ROLE-[0-9]+$	yes	
role.role	string	yes	the role-type as the decided ledgers name it. Never a person or a position
role.children	array of ^CHILD-[A-Za-z0-9-]+$	yes (may be empty)	the children whose model names it
role.decides	array of {decision, child, element_id}	yes (may be empty)	from decision-domain entries and decisions on crossings
role.joint_with	array of ^ROLE-	yes (may be empty)	where its ledgers declare a decision held jointly

Field	Type	Req	Meaning
role.accountable_for	array of {what, child (optional), element_id}	yes (may be empty)	the capabilities its ledgers give it; the obligations the decided Design Brief gives it
role.appears_in	array of {part, element_id, child (optional)}, minItems 1	yes	every place a part names it
role.sends, role.receives	array of {ref (^HO-[0-9]+$ or ^CPR-[0-9]+$), counterpart (a role-type, or a counterparty class), what_crosses}	yes (may be empty)	from the hand-offs and counterparty registers
role.provenance	enum stated, inferred, unresolved, assumed	yes	the weakest of its basis entries; never upgraded
role.basis	array of ledger and register entry IDs, minItems 1	yes	
role.carried_from	^TOMR-[a-f0-9]{64}$	no	when it comes from a carried child
role.decision_ref	source reference	after T8	
findings	array of role finding	yes (may be empty)	
finding.finding_id	^RF-[0-9]+$	yes	
finding.kind	enum one role, two names, one decision, two roles, decision with no role, hand-off with no role at one end, other	yes	other only when raised at RP6
finding.raised_by	enum code, review; with review_finding (^F-[0-9]+$) when review	yes	
finding.roles	array of ^ROLE-	yes (may be empty)	

Field	Type	Req	Meaning
finding.refs	array of element IDs, HO-n or CPR-n, minItems 1	yes	
finding.routed_to	enum decision row, Design, evidence	yes	never merged or settled by choice
finding.routed_ref	NR-..., or S5/<child>#<element>	yes	
finding.status	enum open, resolved upstream, decided, carried	yes	T6 needs none open
finding.resolution_ref	string	unless open	

The code rules (fixed; rules_version).

1. **Inputs.** The authoritative entries of the decided design ledgers of the slice and of carried children (their role fields, CARRIED-RULES.md §8.2); the hand-offs register (role_a, role_b, decision_on_crossing); the counterparty register (crossings[].role); the obligations of the decided Design Brief (their owner role).
2. **One entry per name.** Names are compared after normalisation **N1**: case folded, spacing collapsed, punctuation removed. Two names that differ after N1 are two entries. Code never merges them.
3. **Findings, by fixed rules.**
   - one role, two names: two entries whose names are equal after **N2** (N1; then the words "of", "the" and "and" dropped, a final plural "s" dropped, and the words sorted).
   - one decision, two roles: two decision-domain entries of one child whose decisions are equal after N1 and whose roles differ, with no joint holding declared.
   - decision with no role: a decision-domain entry, or a decision_on_crossing, whose role is empty or unknown.

- hand-off with no role at one end: a hand-offs register entry whose role_a or role_b is empty or unknown, or names a role that its child's ledger does not hold.
- RP6 may raise any other conflict (other), and any two names the fixed rules miss.

4. **Routing.**
   - one role, two names → a non-delegable decision row: same role, different roles or don't know (H-33). Only same role joins two names, and the join is recorded on the entry.
   - one decision, two roles → Design: the child re-runs S5 from step 4 (the Ledger Reviser), and RP5 runs again; or, where two sources disagree on who holds the decision, a contradiction row (Annex B.2).
   - decision with no role and hand-off with no role at one end → evidence (a needs request for the fact-type, for example EX-04) or Design. Where the evidence is declared absent, the finding is carried: the gap is an unknown with its cost, the walk meets it (W3), and the Decision Package lists it in block D.
5. **Not a model of the whole.** It adds no role, decision, capability or body. It lists what the children's decided models and the registers already hold (D6).
6. **Re-compiled** after any correction. At T9, code re-compiles the role entries from the frozen parts and the decision references; they must equal the frozen part's.

Done when. the register compiles from its inputs, and compiling again gives the same bytes; no role finding is open (T6); RP6 is settled; every entry is decided (T8); it re-compiles at T9.

E.18 The difference register (run record; S1, and S5 per child)

One per two-author output. Written by the compare code step; the reconciler fills each disposition.

Field	Type	Req	Meaning
register_id	^DIFR-[A-Za-z0-9-]+$	yes	
run_id	^RUN-	yes	
stage	enum S1, S5	yes	
child	^CHILD-	for S5	
drafts	array of exactly 2 {author_seat (Cell code), family, draft_sha256, pack_version}	yes	both locked before the compare step ran
compare_rules_version	string	yes	the alignment rules of the compare step
reconciler	{seat, family}	yes	the family is one of the two drafts'
differences	array of difference	yes (may be empty)	
difference.difference_id	^DIF-[0-9]+$	yes	
difference.item	string	yes	the section and item the code aligned (S1: the frame section and row; S5: the element type and claim)
difference.kind	enum wording only, label or answer, one side only, conflicting	yes	label or answer: a label, provenance, confidence, standing, Gate A answer or floor result
difference.material	boolean	yes	Annex G.4
difference.side_a, side_b	quote of a locked draft, or null	yes	null where that draft lacks the item
difference.alignment	enum code, split by reconciler, joined by reconciler; with alignment_reason (quote) unless code	yes	

Field	Type	Req	Meaning
difference.disposition	enum either wording, weaker side, kept: its atoms state it, dropped: its atoms do not state it, settled on the sources, decision row, needs request, held open	yes	Annex G.5
difference.basis	array of quote	for kept..., dropped..., settled on the sources	
difference.request_id	^NR-[A-Za-z0-9-]+$	for decision row, needs request	
difference.finding_id	^F-[0-9]+$	for held open	the MINOR finding code raised at the review point
difference.reconciled_item	string, or null	yes	its place in the reconciled version
counts	object: one integer per disposition	yes	

E.19 The reference-model catalogue, as this AIOC reads it (contracts/reference-model-catalogue.schema.json, $id urn:49gaion:library:tom:reference-model-catalogue:v0.4)

A Library item of 49GAION, proposed in upstream/REFERENCE-MODEL-CATALOGUE.md. Read at the version and hash pinned in contracts/PINS.json; never a source; never in the Input Pack. additionalProperties: false at every level.

Field	Type	Req	Meaning
catalogue	const 49GAION Reference-Model Catalogue	yes	

Field	Type	Req	Meaning
version	^v[0-9]+\.[0-9]+\.[0-9]+$	yes	
status	enum draft, released	yes	
rules	array of string, minItems 1	yes	the use rules every reader keeps
patterns	array, minItems 1	yes	
pattern.pattern_id	^RMP-[A-Z0-9-]{2,24}$	yes	stable; never reused
pattern.name	string, 1 to 80 characters	yes	plain; the only part of a pattern that reaches a release
pattern.summary	string	yes	
pattern.typical_capabilities	array of string, minItems 1	yes	
pattern.decision_rights	array of {decision, role_type}, minItems 1	yes	by role-type, never a person
pattern.typical_hand_offs	array of {from, to, what_crosses}, minItems 1	yes	
pattern.fits_when, does_not_fit_when, known_risks	arrays of string, minItems 1	yes	
pattern.never_evidence	const "This pattern is never evidence for an organisation."	yes	

A pattern reference in a design ledger: {pattern_id, catalogue_version, suggested}; suggested is a string naming what the pattern suggested for this element.

The pin (contracts/PINS.json, key reference_model_catalogue):

```
"reference_model_catalogue": {
  "repository": "49GAION-protocol",
  "path": "library/reference-models/reference-model-catalogue.json",
```

```
  "status": "not in the Library",
  "version": null,
  "commit": null,
  "sha256": null,
  "note": "Proposed in upstream/REFERENCE-MODEL-CATALOGUE.md. Until 49GAION holds it and this pin names its version, commit and SHA-256, every run declares the catalogue unavailable (blueprint §5.5). Taking a version into use is H-34."
}
```


E.20 The release notice (contracts/release-notice.schema.json, $id urn:49gaion:library:tom:release-notice:v0.5)

Sent to the next stage on every filing of a decided release (§5.9 step 8a). It points; it never carries release text. **New** in v0.5.0. It is this AIOC's specialisation of the Library's release notice contract, proposed to 49GAION: the same fields, with this AIOC's patterns and values.

Field	Type	Req	Meaning
notice_id	^TOMN-[a-f0-9]{64}$	yes	TOMN- + the SHA-256 of the notice without this field
from	[ORG]AIOC-TOM	yes	this AIOC
release	{release_id ^TOMR-[a-f0-9]{64}$, version, manifest_sha256}	yes	the release it announces
rests_on	empty array	yes	the decided releases this release was designed from: none, since a TOM Release rests on sources only
filed_ref	{source_id, version, sha256}	yes	the release's filed copy in Source Lifecycle, as the filing receipt names it

Field	Type	Req	Meaning
final_status	ACCEPT · ACCEPT WITH CORRECTIONS	yes	the decided final status; no notice is sent for any other
final_status_ref	string	yes	the decision reference in Source Lifecycle
supersedes	^TOMR-[a-f0-9]{64}$ or null	yes	the release it supersedes, if any
answers	array of ^[A-Z]{2,8}R-[0-9]+$	yes	the returns this release answers (§5.10); may be empty
to	[ORG]AIOC-PPD	yes	the next stage
sent_at	date-time	yes	

No other field. A notice holds no element text, no atom text and no name of a person.

E.21 The returns log (run record of this AIOC, across runs)

One entry per return, appended at each of its versions; never edited. **New** in v0.5.0.

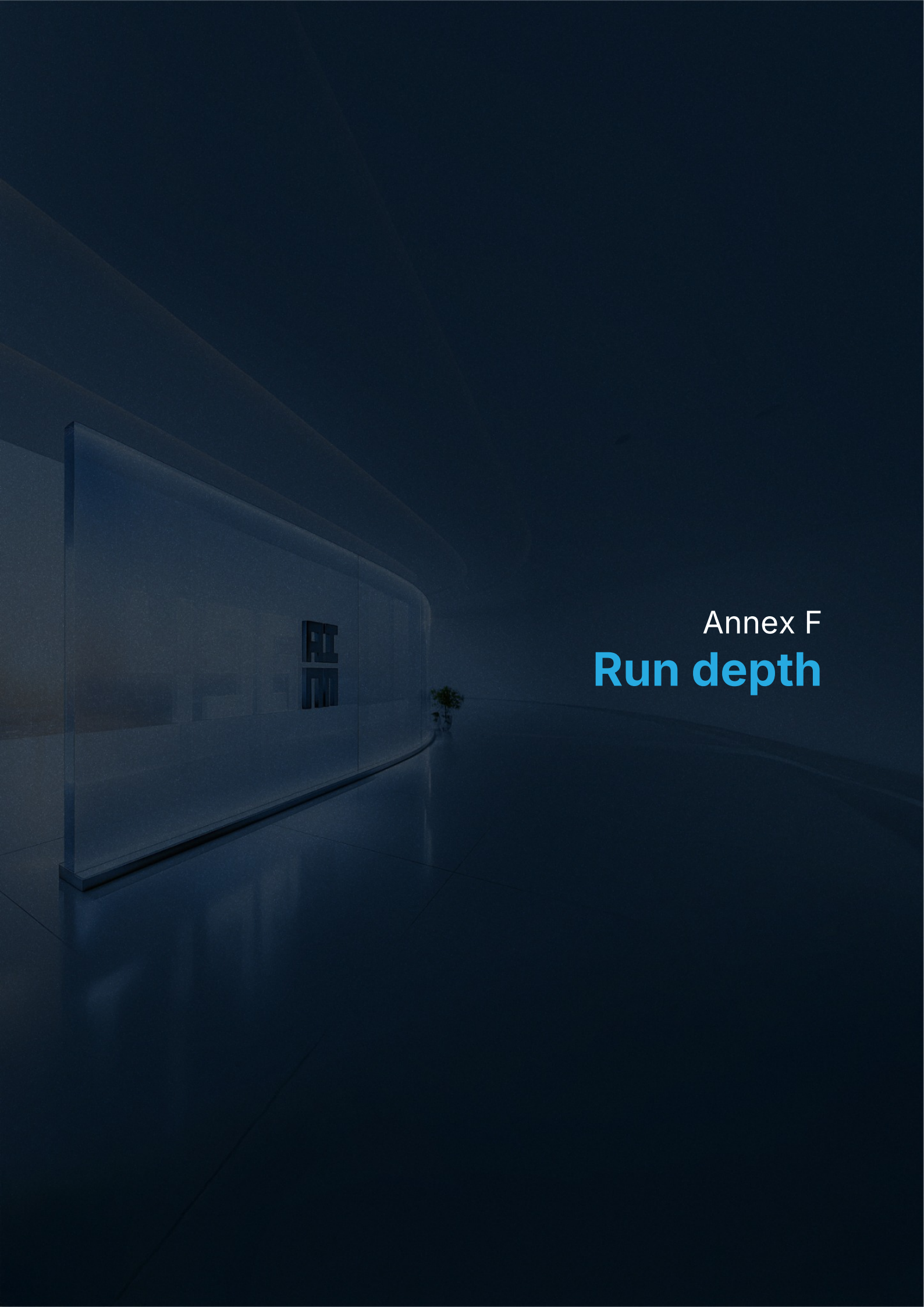
Field	Type	Req	Meaning
return_id	^[A-Z]{2,8}R-[0-9]+$	yes	as sent
sender	[ORG]AIOC-PPD · [ORG]AIOC-DI	yes	the connection it came on
return_sha256	SHA-256	yes	of the return at this version
round	integer ≥ 1	yes	as sent, or as opened by this AIOC's answer

Field	Type	Req	Meaning
deadline	date-time	yes	the end of the round, from return_round_deadline
admission	admitted · refused: <reason>	yes	by §5.10, limit 1
status	the contract's status	yes	
answered_by	release ID or decision reference	for an end other than withdrawn	
carried_forward_on	known defect · decided assumption	for set aside by decision	
decision_refs	array of strings	yes	every decision recorded on it
escalated_at	date-time or null	yes	§5.10, limit 6
classification	defect of this AIOC · normal discovery · waiting	yes	by the TOM Design Lead (H-38)

E.22 The return brief (made by code; the explainer of the return's rows)

Field	Type	Req	Meaning
return_id, round	as the return	yes	
kind	the contract's kind	yes	
elements	array of {part, element_id, plain name}	yes	each element the return names, with its name in the release
missing	string	for cannot use	as the return states it
evidence	array of references	yes	as the return states it; never text
rows	three rows (open a continuation run; answer by decision; set aside on a stated basis), or two after escalation	yes	one decision per row

Nothing is written by hand into a brief; the release lint runs on it.



Annex F

Run depth

New in v0.4.0. In plain words. The run depth says how fine a model the organisation needs: light, standard or deep. It changes how much is designed and checked above the floors. It never removes a stage, a capability, a gate or a check, never lowers a review, and never changes how the organisation is asked.

F.1 Who sets it.

1. **The Requesting Authority** states the depth as its need, in the approved Design Request (DRQ-01 field 9, docs/ASKING-CATALOGUE.md §3a). The socket reads it from that field's atoms at P0 (Annex E.1, design_request.depth). Where the field is empty, the depth is standard.
2. **The Cell carrying [ORG]HOAC-TM** limits it for cost, with the slot run_depth_permitted (default standard; H-31). A depth above the permitted one fails T0 with depth not permitted (Annex C.2, condition (i)). It is never lowered silently: the Requesting Authority may state a lower depth in a revised Design Request, or the TM Cell may raise the permission.
3. **Nobody else.** The TOM Design Lead neither sets nor changes it. Nothing received through a human port sets it.
4. **Fixed for the run.** It is shown in the run preview, on the Panel and in the manifest (depth). A continuation run states its own. A carried child keeps the grain at which it was decided, and the Decision Package says so where the depths differ.

F.2 What it changes, and the floors it never crosses.

	light	standard	deep	Floor at every depth
Capability levels in the Blueprint (S5)	level 1 only	as far as the child's evidence tells structures apart (CARRIED-RULES.md §8.1 rule 11)	as standard; and for every major or core capability both authors attempt one level finer: authoritative where the evidence tells it apart, otherwise a non-authoritative candidate with the fact-type that would settle it	architecture level (rule 2); the coarsest defensible grain (rule 11): the evidence may stop a grain earlier, never later
Value streams (S5)	phases at level-1 capabilities	capability-level phases (rule 2)	each stream of a major or core capability broken to phases at the finest capability level designed	capability-level phases; never procedure
Situations walked (S7)	the floor set: W1 for every declared stream, every mandatory negative, the blind pack at its 30% share and the library at its 60% share	the scored set (CARRIED-RULES.md §10.5 item 6)	the scored set, plus one W3 per decision domain and one W4 per declared crossing, as additive situations	the mix shares; W1 per stream; the mandatory negatives; a shortfall declared
Lenses per frame, per author (S1)	at most 3	at most lenses_max	at most 8	at least 2, over the two author families (CARRIED-RULES.md §2.7)
Review, at every review point	as configured	as configured	five engines, and three families where the register allows	§5.8 rules 2–4, 12, 16 and 18

At light, an element whose purpose needs a level finer than level 1 is recorded in B9 and in the Decision Package (block D). It is not incomplete-design for that reason alone.

F.3 What it never changes. The nine stages and their gates; every capability, code check and review point; the Review Panel rule; the decision rules of Annex C.4 (rows per pass, one decision per row, non-delegable rows, confirmations, approvals); the needs rules of §5.2; the two authors of Annex G; the labels and the honesty ceiling. No depth asks for observation of real work (H-07). A request that is large in its reach is answered by decomposition, never by depth (CARRIED-RULES.md §6.10 item 2). The review budget (§5.8 rule 16) holds at every depth and never lowers a review.

F.4 Not the earlier depth setting. The case study had a depth setting. The engine chose it itself, from how reversible a request was and how far it reached, and it changed what the person asking had to do: it merged approval steps, cut rows and skipped base passes, and at its deepest it required observation of real work. v0.3.0 dropped it, because the burden of asking is set by leverage, passes and rows (docs/DESIGN-BASIS.md §10). The run depth is a different setting. The Requesting Authority states it, the TM Cell limits it for cost, and it changes only how much is designed and checked above the floors. Asking is the same at every depth.

Done when. The depth is recorded at T0 with its source; every stage applied the column of its depth; the manifest and the Decision Package state it.



Annex G

Two authors

New in v0.4.0. In plain words. At Frame (S1) and at Design (S5), the work is written twice. Two authors, on engines of two different families, each write it alone, blind to the other. Code compares the two drafts. A reconciler makes one version. It may settle a difference only on the sources, by taking the weaker side, or by putting it to the organisation; otherwise the difference stays open, at the coarser level both drafts share. The review of the stage then checks the result with both drafts and every difference in its packet. Two authors add to the Review Panel rule; they replace none of it.

G.1 Where and who.

Stage	Author A	Author B	Reconciler	What is written twice
S1 Frame	Request Analyst ([ORG]CELL-TRA)	Second Request Analyst ([ORG]CELL-TSRA)	Frame Reconciler ([ORG]CELL-TFR)	§5.1 steps 1–9: the frame with its Gate A record, the party register, the do-not-ask and do-not-use-as-fact lists, the needs candidates
S5 Design, per child	Designer ([ORG]CELL-TD)	Second Designer ([ORG]CELL-TSD)	Ledger Reviser ([ORG]CELL-TLR), before it revises	§5.5 step 2: the design ledger

S3 and every other stage keep one author.

G.2 Blind.

1. Each author receives the same inputs, at the same pack version, in its own fresh contexts. At S1 each author makes its own read plan, rounds with the Source Proxy, lens pass, Working Context and needs candidates.
2. Neither author's packet holds anything of the other's work. Each calls under its own credential.
3. Each draft is locked by hash before the compare step runs.

G.3 Families.

1. The two authors are bound to engines of two different families: the run's two author families. Each author's Source Proxy rounds run on that author's family. The lens pass spreads over the two author families (CARRIED-RULES.md §2.7).
2. The reconciler is bound to an engine of one of the two author families, taking them in turn from one job to the next.
3. So an output with two authors has exactly two author families, and in normal mode neither sits on its review (§5.8 rule 3). A full review needs at least three engines of at least two other families: the engine register needs at least four families (H-05). The lens pass already needs the two author families, so two authors at S1 add no family.

G.4 Compare (code). The compare code step aligns the two locked drafts item by item: by section and type, and by the atoms each item cites, under fixed, versioned rules. It writes the difference register (Annex E.18): every item on which the drafts differ, its kind, and whether it is material. A difference is **material** when it touches:

- at S1: a load-bearing item (Annex B.3), a label, a standing, a Gate A answer, the floor, the request kind, or the party register;
- at S5: an authoritative element (its presence, claim, provenance, confidence, basis or class), an interface entry, the entity type used (A1), the run status (A6), or a pattern reference.

Every other difference (wording, order, a narrative cell with the same linked entries) is not material.

G.5 Reconcile. The reconciler works in a fresh context, with only the two locked drafts, the difference register, the atoms the drafts cite and the rules. It may correct the code's alignment only with a quoted reason. For each difference, in this order:

1. **Not material:** either wording is kept.
2. **A label, provenance, confidence, standing, Gate A answer or floor result:** the weaker side, always. Reconciliation never raises anything.
3. **On one side only:** kept if its cited atoms state it exactly; otherwise dropped. A party in one party register only is kept, with its quotes. A needs candidate

on one side only is kept; the Chief of Staff drops duplicates, and the leverage test applies (§5.2).

4. **In conflict, and the cited atoms settle it** (one side says exactly what its atoms say; the other adds, drops or alters): the supported side, with the quotes.
5. **In conflict, and the atoms do not settle it:**
   - at S1, where it is what the organisation asks (scope, an exclusion, the roster, an outcome): a non-delegable decision row, answered reading A, reading B, both readings or don't know (H-20). Where two sources disagree: a contradiction row (Annex B.2). Anything else: an open item carrying both readings, labelled inferred, contested-basis;
   - at S5: never a design question (docs/ASKING-CATALOGUE.md §5). Where a fact or an intent the sources lack would settle it, that fact-type is asked as a needs request, and the job waits as for incomplete-design (CARRIED-RULES.md §8.1 rule 12). Otherwise it is **held open**: the authoritative ledger keeps only what both drafts share, at the coarser grain, and each finer variant goes to B5 as a non-authoritative candidate, with what would settle it in B9 (rule 11).
6. The reconciler adds nothing that neither draft holds, never chooses by preference, writes no verdict, and records each disposition with its reason and quotes.

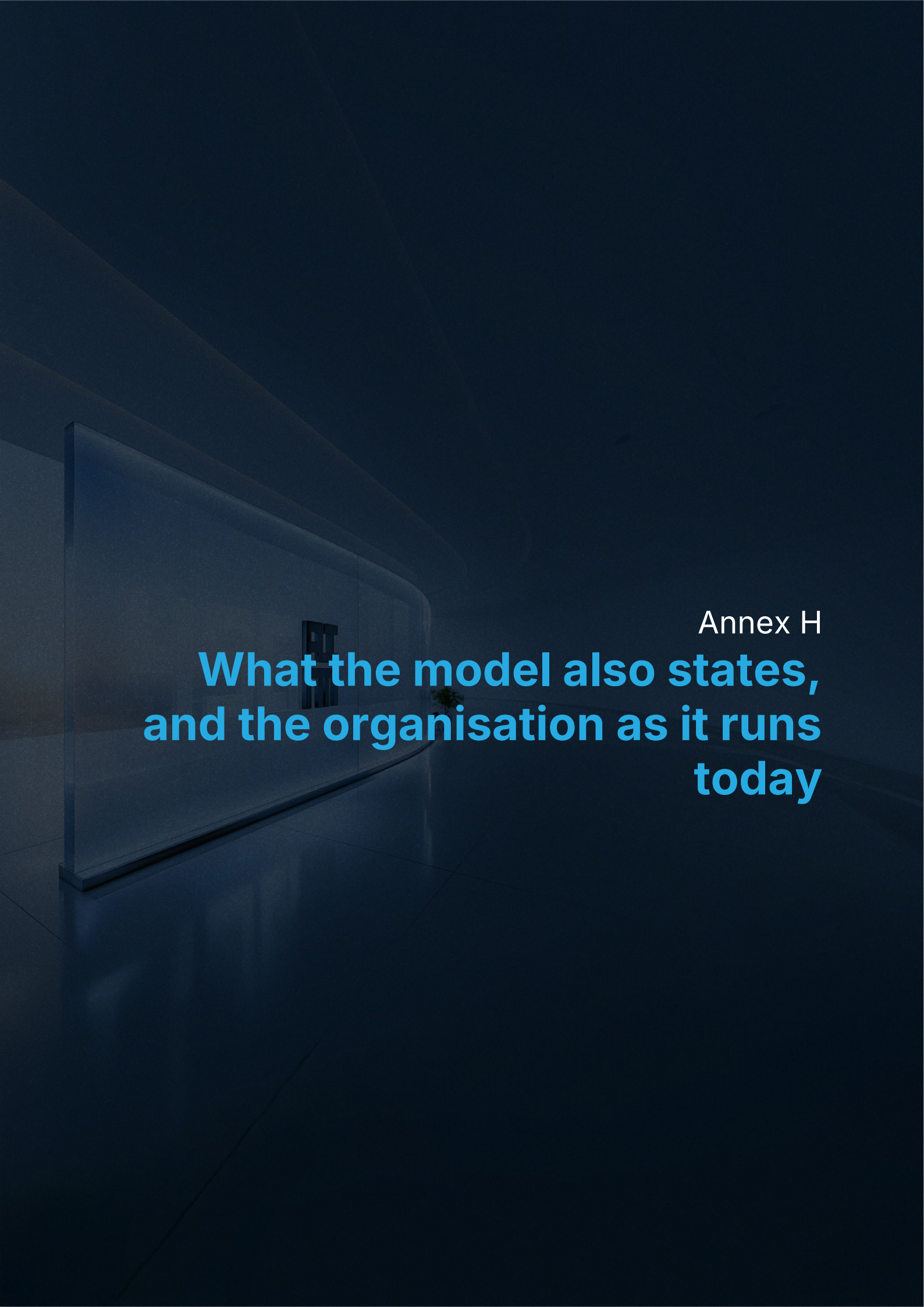
G.6 What reaches review. The review points of the stage (RP1b and RP1 at S1; RP5 at S5) receive both locked drafts and the difference register in their packets. Each material difference is a checkable item (§5.8 rule 8). Code raises one MINOR finding on each difference held open, as it does on an item not covered. A wrong disposition is a MATERIAL finding at that point (§5.8, review points).

G.7 When the register cannot supply two authors. Never a silent single author.

1. **S1 always has two authors.** If, at P0's highest confidentiality level, author engines of two families are not eligible, T0 fails with no second author (Annex C.2, condition (j)). The lens pass already needs those two families.

2. **Two authors never lower a review.** Where two authors at S5 would bring its review below the level one author allows, S5 runs with one author only if the slot single_author_permitted is yes (the Cell carrying [ORG]HOAC-TM; default no; H-32) and the TOM Design Lead accepted single authoring at S5 at the run preview. Otherwise T0 fails with no second author.
3. **Declared.** Single authoring is shown in the run preview, on the Panel, and in the quality checks of S5 with the fixed sentence: "The work at this stage was written once, not twice independently." The Requesting Authority is told, not asked.
4. **Mid-run.** If either author family can no longer be bound at the pack's highest level, the run holds as held: engine ceiling (Annex C.1) until it can be, or until the Requesting Authority withdraws the run. A run never goes on with one author silently.

Done when. For each two-author output: both drafts locked before the compare step; the difference register complete, with every material difference disposed under G.5; its decision rows and needs requests sent through the Chief of Staff (§5.2); the stage's reviews settled as §5.8 requires.



Annex H

What the model also states, and the organisation as it runs today

New in v0.6.0 (decisions/DR-0006). **In plain words.** The model now also states the organisation's strategy on one page, how each capability is delivered and how mature it is, which value streams are critical, the path each outside party takes through the organisation, the value the change is meant to bring, and the priority of each element. It can also read records of how the work runs today, never about a person, and where there are none it says so and goes on.

Nine additions, each placed in a stage that already exists. They add no stage, gate, review point or seat, and they remove or weaken nothing. Every rule of this blueprint applies to them: citation or refusal, labels (Annex B), one decision per row, and roles and positions only. **This AIOC never writes a goal, a target, a level or a figure of its own.** Each is stated by a held source, decided by the organisation, or shown as not given, unknown or not set.

#	Addition	Stage	Where it is kept	Who decides
H.1	Strategy frame	S1	release part strategy frame	the Requesting Authority, one row; purpose and values, the Cell carrying [ORG]HOAC-GOV
H.2	Delivery model and make-or-buy, per capability	S5; checked at S6	design ledger	the Requesting Authority, with the capability map row
H.3	Critical value streams	S5	design ledger	the Requesting Authority, with the value stream row
H.4	Maturity per capability, now and target	S5	design ledger	now: a held source; target: a held source or the Requesting Authority
H.5	Outside parties' journeys	S6	release part outside-party journeys	the Requesting Authority, with the counterparty class row

#	Addition	Stage	Where it is kept	Who decides
H.6	Value case	S8	release part value case	the Requesting Authority, one row
H.7	Priority per element	S8	each element's decision (decisions[].decision.priority)	the Requesting Authority, with each element row
H.8	Records of the work as it runs today, and the activity profile	S0, S1	release part current-state activity profile	the organisation admits each export in Source Lifecycle (H-07, H-40)
H.9	The organisation dictionary	S1	read as a held source	—

H.1 The strategy frame (S1)

1. **What it holds.** Six elements: the purpose; the values; the long-range target; the three-year picture; the one-year goals, each with its measure; the critical numbers, each with its definition. Field list: H.10.1.
2. **Where each element comes from.** From held sources, cited, with their labels. The authors of the frame never write an element and never derive one from other material: what the organisation wants is never inferred (docs/CARRIED-RULES.md §1.2 item 4).
3. **Asked once.** Where the sources lack any element, the first pass of S1 carries one explainer, EX-10 "Strategy on one page" (docs/ASKING-CATALOGUE.md §3), naming only the elements missing. Its leverage proof is standing: the priority of every element (H.7), the value case (H.6) and the next stage's objectives start from it. The Chief of Staff of [ORG]AIOC-SL chooses who answers; the answer is recorded there as a source and admitted like any other (organisation-stated).
4. **Purpose and values.** Where a source in force states them, they are cited and not asked again. Where none does, their row is the listed type

set-purpose-or-values (HF-04), signed by the Cell carrying [ORG]HOAC-GOV at the gateway (§5.9, Purpose and values; H-03).

5. **Decided.** The frame decision pass (§5.1 step 12) carries one non-delegable row for the strategy frame as a whole (ACCEPT, CORRECT, REJECT), which names each element not given. T1 needs it decided. A REJECT is asked again through a new EX-10, within decision_passes_max.
6. **Never blocking.** An element still missing after its answer (declared absent, declined or dont know) is not given. It is shown in the part and in block A of the Decision Package (evidence gaps), and the run goes on. Nothing fills it.
7. **Released.** The decided strategy frame is the release part strategy frame. The Design Brief points to it (docs/CARRIED-RULES.md §4.1, section 22).

H.2 Delivery model and make-or-buy (S5)

1. **Delivery model.** Each capability of a child's capability map (A2) carries delivery_model: in-house, shared service, outsourced, partner or not decided. It says how the capability is delivered, never by whom: no provider is named, and a provider outside the organisation is a counterparty marker (§5.6).
2. **Make or buy**, computed by code, never written by an author: make for in-house and shared service; buy for outsourced and partner; not decided otherwise.
3. **A design choice, with its basis.** The delivery model carries its provenance, confidence and basis like any element (docs/CARRIED-RULES.md §8.2): stated where an atom states it; inferred where the design chooses it on cited atoms; not decided where the child's own evidence cannot tell the options apart (§8.1 rule 11, the coarsest defensible grain). A pattern may suggest it and never supports it alone (Annex B.5).
4. **Carried by a crossing.** A capability that is shared service, outsourced or partner names, in delivered_through, the crossings of its child's register that carry it. One that names none is an undeclared-crossing finding, which returns the package to Entity Packaging (§5.5). At S6, each such crossing is paired as a hand-off, or appears in the counterparty register, or is a deferred

boundary of the class of service providers (docs/CARRIED-RULES.md §9.2 item 1); otherwise it is a seam finding (H.11 check 2).

5. **Decided** with the capability map row (§5.9).

H.3 Critical value streams (S5)

1. Each value stream (A3) carries critical: yes, no or not decided. A yes names at least one reason from a closed list, revenue, customer, legal or regulatory, safety or continuity, and each reason cites the atoms that support it.
2. It is the authors' proposal, with provenance like any element. A stream is never critical on a pattern alone.
3. **Decided** with the value stream row (§5.9). A critical stream sets the priority of the elements on it (H.7).

H.4 Maturity (S5)

1. Each capability carries maturity, now and target, each a level of one closed scale: 1 ad hoc, 2 repeatable, 3 defined, 4 measured, 5 improving. The scale is a method, never evidence.
2. **Now** is stated by a held source, such as an assessment the organisation made and recorded, cited, with its label and its date; otherwise unknown. No author assesses a level (docs/CARRIED-RULES.md §8.1 rule 10: no forensic discovery of the present state).
3. **Target** is stated by a held source, or decided by the Requesting Authority in a correction of the capability map row (§5.9, Corrections: a design choice, applied as a decision); otherwise not set.
4. Where a level would change a design choice (the leverage test, §5.1 step 9), it is asked through EX-11 "How mature each capability is", once per child.
5. Maturity is of a capability. It is never of a role, a position or a person.

H.5 Outside parties' journeys (S6)

1. For each entry of the counterparty register (Annex E.5), the Hand-off Analyst records its journey, JNY-n: the moments where that party meets the organisation, in order. Each moment is a crossing already in that entry, with the value stream and the phase it meets on the child's side. Field list: H.10.4.
2. It adds no crossing and designs nothing on the party's side (docs/CARRIED-RULES.md §8.1 rule 9). The order is as the sources state it, cited, or order not stated.
3. A moment the sources describe that no crossing carries is a seam finding (missing counterpart), routed back like any other (§5.6). A crossing that meets no designed phase reads phase not designed.
4. Code checks every moment and every phase (H.11 check 4). The journey is put to decision with its counterparty class row (§5.9). It is the release part outside-party journeys.

H.6 The value case (S8)

1. **One entry per outcome** of the decided frame (Design Brief section 6): the change, as the outcome states it; its measures, which are the criteria decided for it at S1 or S3 and the critical numbers of the strategy frame it moves; for each measure, its baseline (atoms, with their date, or unknown) and its target (an atom or a decision reference, or not set); the elements it rests on; and the assumptions it rests on (Design Brief section 9), each with what breaks if it is false. Field list: H.10.5.
2. **Compiled by code** from the decided parts and the decision references, by the Chief of Staff's value-case code step. It re-compiles byte-identical at T9. It holds no figure that is not an atom or a decision, and no estimate of this AIOC.
3. **Decided** as one non-delegable row (§5.9). A correction may set a target; it is applied as a decision.
4. It is the release part value case. The stage that carries the model into the organisation reads it as the start of its register of value (H.12).

H.7 Priority (S8)

1. **Proposed by code**, never by an author. Each element row (§5.9) carries a proposed priority, by a fixed rule:
   - must: the element lies on a critical value stream (H.3), carries an obligation of the organisation (Design Brief section 12), or holds an item of the legal and regulatory perimeter;
   - should: an outcome of the frame, or a one-year goal of the strategy frame, traces to it (the element's basis cites an atom that the outcome or the goal cites);
   - could: any other element.
2. **Decided** with the element. ACCEPT keeps the proposed priority; CORRECT may change it. The decided priority is recorded on the decision (decisions[].decision.priority). An element row decided ACCEPT or CORRECT without a priority fails T8.
3. Priority orders the next stages' work. It never removes an element and never changes a gate. A carried element keeps its decided priority.

H.8 Records of the work as it runs today, and the activity profile (S0, S1)

1. **What it is.** Records of the work as it runs today, exported from the organisation's own systems (work queues, case logs, workflow and time records), aggregated by role-type and by process. Never about a person. This AIOC never watches work: it reads only what an approved export states. The label observed stays unused (Annex B.1).
2. **How it enters.** As a source of origin kind export, through the same socket and adapter, labelled source-backed as of its export date (Annex B.1a). It is asked, where the leverage test passes, through the export request (docs/ASKING-CATALOGUE.md §4.3), which names the aggregation: by role-type and by process, with no field that says who did the work.
3. **The projection is the organisation's.** Before such an export is recorded in Source Lifecycle, the organisation's own mechanism removes who did the

work, and any line that would still point to whoever did it; the seat that holds data protection (HF-17) records an impact assessment of the export and its decision on it (H-40; docs/INTAKE-WITH-SOURCE-LIFECYCLE.md, SL-A13). This AIOC never sees the records before that projection.

4. **Refused.** An export whose records still carry a field that says who did the work is refused at the socket, person-level records (§4.3; Annex E.1; Annex E.2), and nothing of it is kept.
5. **The activity profile.** At S1, before the authors start, the Chief of Staff runs the activity-profile code step on the pack version named: per role-type and per process, as the admitted exports name them, the share of time or the volume each export states, with its date and its atoms. Field list: H.10.6. It describes today; it designs nothing and sets no workload. Both authors read it as present state (§5.1 step 5), and RP1 reviews it. It is the release part current-state activity profile.
6. **None.** Where no such export is admitted, the part reads no observation, and the run goes on. Nothing is inferred in its place.
7. **Decided by.** GOV directed this input (decisions/DR-0006; §13, H-07). Each export is admitted only through its own HF-17 decision (H-40).

H.9 The organisation dictionary (S1)

Source Lifecycle serves, by reference, the organisation's dictionary: its own terms, each with its definition and its source. Where it is served, the read plan opens it (§5.1 step 1), and it is a held source like any other. In reader text, a term the dictionary defines is used in that meaning. A source that uses the term differently raises a candidate contradiction (Annex B.2). The dictionary decides nothing and is never a precedence rule.

H.10 Field lists

"Element" below means {atoms, label}, where atoms holds at least one atom ID; or {not_given: {needs_ref, outcome}}, where outcome is declared absent, declined or dont know.

H.10.1 Strategy frame (release part strategy frame; made at S1)

Field	Type	Req	Meaning
purpose	element	yes	
values	array of element, minItems 1	yes	one not given entry where none is stated
long_range_target	element	yes	
three_year_picture	element	yes	
one_year_goals	array of {goal: element, measure: element}, minItems 1	yes	one not given goal where none is stated
critical_numbers	array of {name: element, definition: element}, minItems 1	yes	as above
purpose_and_values_ref	source reference	after T1	the source in force, or the signed set-purpose-or-values decision
decision_ref	source reference	after T1	the strategy frame row

H.10.2 Design ledger, capability (adds to docs/CARRIED-RULES.md §8.2)

Field	Type	Req	Meaning
delivery_model	enum in-house, shared service, outsourced, partner, not decided	yes	with the element's provenance, confidence and basis
make_or_buy	enum make, buy, not decided	yes	by code (H.2 item 2)
delivered_through	array of crossing references	unless in-house or not decided	minItems 1
maturity.now	integer 1–5, or unknown	yes	
maturity.now_basis	array of atom IDs, with as_of	unless unknown	
maturity.target	integer 1–5, or not set	yes	

Field	Type	Req	Meaning
maturity.target_basis	atom ID or decision reference	unless not set	

H.10.3 Design ledger, value stream

Field	Type	Req	Meaning
critical	enum yes, no, not decided	yes	
critical_reasons	array of {reason, basis}; reason enum revenue, customer, legal or regulatory, safety, continuity; basis atom IDs, minItems 1	when yes, minItems 1	

H.10.4 Outside-party journey (release part outside-party journeys; made at S6)

Field	Type	Req	Meaning
journey_id	^JNY-[0-9]+$	yes	
counterparty_id	^CPR-[0-9]+$	yes	the entry of the counterparty register
moments	array, minItems 1, of {order, crossing: {child, crossing_ref}, phase, basis}; order an integer or order not stated; phase {child, stream, phase} or phase not designed	yes	each crossing is one of that entry's crossings
seam_refs	array of ^SF-[0-9]+$	no	moments with no crossing
decision_ref	source reference	after T8	its counterparty class row

H.10.5 Value case entry (release part value case; made by code at S8)

Field	Type	Req	Meaning
entry_id	^VC-[0-9]+$	yes	
outcome_ref	reference to Design Brief section 6	yes	
measures	array, minItems 1, of {measure_ref, baseline, target}; measure_ref a criterion or a critical number; baseline {atoms, as_of} or unknown; target an atom ID or a decision reference, or not set	yes	
elements	array of element IDs, minItems 1	yes	
assumptions	array of references to Design Brief section 9	yes, may be empty	
decision_ref	source reference	after T8	

H.10.6 Current-state activity profile (release part current-state activity profile; made by code at S1)

Field	Type	Req	Meaning
state	enum profile, no observation	yes	
exports	array of {source_ref, as_of}	when profile, minItems 1	
lines	array of {role_type, process, measure, value, atoms}; measure enum share of time, volume	when profile, minItems 1	role_type and process as the export names them; value as the export states it

None of these lists holds a field about a person.

H.11 The checks by code

1. **S1.** Every element of the strategy frame has its atoms, or is not given with its needs reference. The activity profile re-compiles from the pack version named, or reads no observation (T1).
2. **S5 and S6.** Every capability has a delivery_model in the closed list and the make_or_buy that follows from it; every shared service, outsourced or partner capability names its crossings (T5); at S6 each such crossing is paired, or in the counterparty register, or a deferred boundary of the class of service providers (T6). (docs/CARRIED-RULES.md §8.3 item 11.)
3. **S5.** Every critical: yes stream has a reason with its atoms; every maturity.now other than unknown cites atoms; every maturity.target other than not set cites an atom or a decision (T5). (§8.3 item 12.)
4. **S6.** Every moment of a journey is a crossing of its counterparty entry, and every phase is a phase of a decided ledger (T6).
5. **S8.** Every element row decided ACCEPT or CORRECT carries its priority; the proposed priorities re-compute by the rule of H.7 (T8). The value case and the activity profile re-compile byte-identical from the frozen parts and the decision references (T9).
6. **S0.** No admitted export carries person-level records (T0 condition (f)).

H.12 How the stages after it read these

Each reads by reference from Source Lifecycle, as it reads every part.

- **Process and Position Design** starts its objectives from the goals and critical numbers of the strategy frame; designs the critical streams first; gives a capability that is outsourced or partner a hand-off to the outside party, never an in-house process; reads the activity profile as today's workload; and orders its work by priority.
- **Delivery and Implementation** carries the value case as the start of its register of value, and the priorities into its rollout.

Done when. The checks of H.11 pass at their gates, and RP1, RP5, RP6 and RP8 settle with these additions in their packets.

Blueprint · [ORG]AIOC-TOM · Target Operating Model · v0.6.3 · Powered by 49GAION


49G AI
49G HOLDING | 49G AI
www.49g.com