AIOC BLUEPRINT TEMPLATE

The one versioned file from which an AIOC's panel and back office are built. Template version v2.1.1.

Vol: 13 | v2.1.1



49G Ltd.
ADGM, Abu Dhabi, UAE

AIOC Blueprint Template

The one versioned file from which an AIOC's panel and back office are built. Template version v2.1.1.

Release: v2.2.1 · 30 October 2026 · approved by the governing seat (GOV) of 49GAION
Source: 49g-holding/49GAION-aioc · template/AIOC-BLUEPRINT-TEMPLATE.md @ 56cc77b
Precedence: 3 · the blueprint template
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

AIOC Blueprint Template

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

blueprint:

  aioc: "<[ORG]AIOC-INITIALS>"            # e.g. [ORG]AIOC-SL or TenAIOC-SL. Only cores carry a [ORG]AIOC code; it never changes when the AIOC moves.

  parent: "<code of the unit that holds this AIOC, or none>"   # e.g. [ORG]AIOC-OCO, an example only; its place, never part of its code; a top-level AIOC's parent is the Charter of its [ORG]AION

  full_name: "<Full name>"                # e.g. Organization Core Office

  organisation: "<organisation>"          # e.g. Ten

  type: "<foundational | structural | specialised | connector | other>"   # why the AIOC exists; a card field, never part of the name

  org_aion: "<[ORG]AION>"                # the [ORG]AION this AIOC belongs to, e.g. TenAION

  blueprint_version: "<vN.N.N>"

  status: "<draft | in force | replaced>"

  template:

    name: "49GAION AIOC Blueprint Template"

    version: "v2.1.1"

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

  mother:

    name: "<Library blueprint name>, or none yet"

    version: "<version>, or none"

    source: "<location>, or none"

  writing_repository: "<org>aioc-<abbr>"

  code_project: "<code project path>, or none"

  panel_register_id: "<panel register id, or none>"

  capabilities:                           # plain English names only; no codes, no [ORG]AIOC prefix

    standard: ["Chief of Staff", "Office", "Input socket", "Panel"]   # Office only where a position needs more than one Chief of Staff

    functional: ["<name>", "<name>"]
```

Blueprint · <[ORG]AIOC-INITIALS> · <Full name> · <blueprint version>

The 49GAION AIOC Blueprint Template, v2.1.1. Blank: every field is a placeholder; it holds no AIOC's content.

> **What this file is.** This is the one git file that defines an AIOC. It is its blueprint and its panel definition. It lists **every capability** the AIOC has, and each capability is written in **four parts: Definition, Process (SOP), Role card, Fixed test**. The AIOC's Panel and its Back office are both **built from this one file**. A change to the AIOC is a **tested git commit** to this file; the Panel then rebuilds from that commit, so the blueprint never drifts from what runs (docs/WHAT-IS-AN-AIOC.md §2). Where this template names [ORG]AIOC-OCO, it means the hub of the operation: the Top Management Group, or a separate [ORG]AIOC-OCO where the organisation has one. The Panel is the organisation's own interface, written in code; the Back office, the AI chat workspace, keeps it. > > **How to fill it.** > 1. Copy this file into the AIOC's writing repository as blueprint/<AIOC>-BLUEPRINT.md. Record in the metadata block the **template version and the commit** it was copied from (each AIOC records this in its own repository; 49GAION-aioc keeps no copy of any AIOC's blueprint), and the **mother blueprint and version** it was customised from, or "none yet". > 2. Fill every field. If a field does not apply, write "none" and say why. > 3. Keep the **standard capabilities** (§4). Adopt their default definitions (docs/STANDARD-CAPABILITIES.md) unless the AIOC differs, and record each difference in §11. > 4. Add one **functional capability** per function the AIOC performs (§5), each in the four parts. > 5. Write for any organisation where you can; the organisation's own names go in the organisation table (§2). > 6. Write positions, never people. No person's name appears anywhere: who holds a position is the organisation's own mechanism, outside the system (DR-0049). > 7. Cite every rule to its source by ID and version. Never invent a fact, a version, a score or an appointment. NOT_RUN is never PASS. > 8. **Names.** Capabilities have plain, recognisable English names — Chief of Staff, Office, Input socket, Panel, and the AIOC's own (for example Service Desk). They carry **no code and no branding**. Branded names exist only at the 49GAION level: 49GAION for the framework, [ORG]AION for its product, [ORG]AIOC-... for cores (hub standards/49GAION-NAMING-STANDARD.md). The retired codes listed there are never used.

Sources: DR-0041.



Section 00

Identity and lineage

Field	Value
Code	<[ORG]AIOC-INITIALS>, checked for collisions (Naming Standard §5)
Full name	<Full name>, written beside the code once
Description	"<code> is <organisation>'s <full name> core, built from <Library blueprint name>"
Kind	AIOC
Parent	the code of the unit that holds this AIOC, or none (Naming Standard R14); a change needs the approval of the human seat over this AIOC
Inner units	the codes of the AIOCs this AIOC holds, each linked to its own card and Blueprint, and of its own Pods and Cells (§17), or "none" (Standard §1.7)
HOAC label	the HOAC label this AIOC carries as a whole, if any ([ORG]HOAC-GOV, [ORG]HOAC-TM or [ORG]HOAC-ACS)
Type	<Foundational / Structural / Specialised / Connector / other (described)>: why this AIOC exists (docs/WHAT-IS-AN-AIOC.md §4).
[ORG]AION	<[ORG]AION>, based on 49GAION Standard <version>
Versioned Definition	this file, at <commit>; the Panel and the Back office are built from it
Organisation	<organisation>

Field	Value
Made from	the Library blueprint <name> <version>, or "none"
Template	49GAION AIOC Blueprint Template <version> at <commit>
Mother	<name, version>, or "none yet". A Library mother is named in the Library register (library/README.md of 49GAION-protocol), pinned by repository, path, version, commit and SHA-256. Where this file is itself a Library mother, it holds no organisation's data, and the organisation's own values live in its records repository (Naming Standard R9)
Blueprint version and status	<vN.N.N> · <draft / in force / replaced>
Accountable seat ([ORG]HOAC-ACS)	the seat, by position code; who holds it is the organisation's own mechanism, outside the system
Reports to	the position or office above it
Writing repository	<org>aioc-<abbr> (+ -records, -records-restricted)
Code project	its code project, or "none"
Panel	its panel's name and register id, or "not yet built"
Back office	its Back office workspace, and the commit it was last built from
Attribution	Powered by 49GAION

Sources: DR-0019.



Section 01

Card (49GAION Operating Protocol §3)

The card declares what this AIOC receives, what it delivers, and in what format; it is the card of 49GAION Operating Protocol §3 (Protocol §4 is the Work Item). It is the only card for this AIOC; the protocol AIOC's map reads it from here. It follows spec/schemas/unit-card.schema.json in 49g-holding/49GAION-protocol.

```
card:
  unit_kind: "AIOC"
  code: "<code>"
  parent: "<code of the holding unit, or none>"
  inner_units: ["<code>"]                # by link; empty if none
  type: "<foundational | structural | specialised | connector | other>"   # require
d; if other, add type_description
  org_aion: "<[ORG]AION>"
  card_version: "<vN.N.N>"
  input_door: "Chief of Staff"           # always; the AIOC's voice to its HOAC se
at (input arrives only through the typed ports and the Input socket)
  wake: { method: "queue", verified: false }   # runtime woken from its queue
  receives:
    - kind: "<a kind of work>"
      format: "<format>"
      acceptance_test: "<test ID and version>"
      decision_needed: false
  delivers:
    - kind: "<a kind of output>"
      format: "<format>"
      usual_receivers: ["<AIOC>"]
  anatomy:
    panel: { health: { tests: true, metrics: ["stuck", "returned", "late"] } }
    back_office: { converses_with_humans: true, hosts: ["Chief of Staff", "Office"]
, protocol_traffic: true }
    definition: { location: "<repository and path of this file>", version: "<commit
 or version>" }
    delivery: { rollback: "one-click" }
    connectors:
```

```
      - { role: "<tool role, e.g. task tracking>", connector: "<connector id>", acc
ess_managed_in: "panel" }   # the Panel shows and requests it; the grant is signed
at the tool gateway as connect-external-port (49GAION/DR-0019)

    gateway: { metering_id: "<per-AIOC metering id on the shared AI gateway>" }

  # With the keys below, the card carries every key the unit card schema requires (
spec/schemas/unit-card.schema.json).

  full_name: "<Full name>"

  version: "<N.N.N>"

  lineage: { id: "<code it derives from>", version: "<N.N.N>", relation: "<original
 | derived | merged | split>" }

  lifecycle_state: "<proposed | designed | built | joined | live | replaced | retir
ed>"   # a state before joined: see Joining the protocol, below

  blueprint_version: "<vN.N.N>"

  template_version: "<vN.N.N>"

  accountable_to: "<code of the human Cell carrying a HOAC label that this AIOC ans
wers to>"   # Standard §1.7, item 12

  top_cell: "<code of exactly one Cell whose parent is this AIOC>"
          # Standard §1.7, item 11

  hoac: "<[ORG]HOAC-ACS | [ORG]HOAC-TM | [ORG]HOAC-GOV>"   # omit if none; with it,
 label_holder (decision_rule withdrawn, DR-0049)

  label_holder: "<code of the human Cell, inside this AIOC, that holds the label>"

  fixed_test: { id: "<test ID>", version: "<vN>", owner: "<code of the human Cell t
hat owns the test>" }

  skills:

    - { id: "<skill-id>", name: "<skill name>", tags: ["<tag>"] }

  ports:                                  # one input and one output of each kind,
plus exactly one stop door (Standard §6.1)

    - { unit: "<code>", port: "ai-in", direction: "input", type: "AI", identificati
on: "technical-key", accepted_content: ["<content>"], authority_limit: "<limit>", a
udit_trail: "<where recorded>", required_tests: ["<test ID and version>"], enforcem
ent: "shared-gateway-outside-unit" }

    - { unit: "<code>", port: "ai-out", direction: "output", type: "AI", identifica
tion: "technical-key", accepted_content: ["<content>"], authority_limit: "<limit>",
 audit_trail: "<where recorded>", required_tests: ["<test ID and version>"], enforc
ement: "shared-gateway-outside-unit" }

    - { unit: "<code>", port: "hu-in", direction: "input", type: "HU", identificati
on: "live-human-sign-in-at-gateway", accepted_content: ["<content>"], authority_lim
it: "<limit>", audit_trail: "<where recorded>", required_tests: ["<test ID and vers
ion>"], enforcement: "shared-gateway-outside-unit" }
```

```
    - { unit: "<code>", port: "hu-out", direction: "output", type: "HU", identifica
tion: "live-human-sign-in-at-gateway", accepted_content: ["<content>"], authority_l
imit: "<limit>", audit_trail: "<where recorded>", required_tests: ["<test ID and ve
rsion>"], enforcement: "shared-gateway-outside-unit" }

    - { unit: "<code>", port: "stop", direction: "input", type: "HU", identificatio
n: "live-human-sign-in-at-gateway", accepted_content: ["stop"], authority_limit: "s
top only", audit_trail: "<where recorded>", required_tests: ["<test ID and version>
"], enforcement: "shared-gateway-outside-unit", stop_door: true }

  configuration:                          # §15; every slot filled, each with who s
et it, when and in which version

    office_accountable_human: { value: "<HOAC seat accountable for OCO>", set_by: "
<seat>", date: "<date>", version: "<vN>" }

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

    discussion_limits: { value: "<limits>", set_by: "<seat>", date: "<date>", versi
on: "<vN>" }

    waiting_deadlines: { value: "<per status>", set_by: "<seat>", date: "<date>", v
ersion: "<vN>" }

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

    chain_depth: { value: "<number>", 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: "<seat>", date: "<date>", v
ersion: "<vN>" }

    unattended_action: "<read-record-triage | act-within-limits>"
```

Joining the protocol. This AIOC joins the 49GAION Operating Protocol only after every input and output claimed on this card passes fixed samples. A claim that fails is removed from the card. Until then the card's lifecycle_state is a state before joined, and no Work Item is routed to it.



Section 02

Organisation table and placement

2.1 Organisation table

The design uses general terms; this table says what each term is in this organisation.

Term in the blueprint	In <organisation>
The Top Office (the only centre of decision: the Office of the root AIOC, the Charter; decisions are signed by human Cells carrying HOAC labels)	
[ORG]AIOC-OCO (every [ORG]HOAC-ACS request about this unit's AIs goes to it)	
The engineering division	
The IT and infrastructure division	
The builder of every tool	
The Library blueprint this core is made from	

2.2 Placement

- **Where it sits:** under whom, beside whom.
- **Its role:** one sentence.
- **Scope:** what it does; what it does not do.
- **Companies it serves:** all, or which.
- **Boundaries with other AIOCs:** each boundary, and who holds what on the other side.

- **Overlap:** each overlap with an existing AIOC and how it is resolved.



Section 03

Positions

Every AIOC, Pod and Cell has one accountable [ORG]HOAC-ACS seat: it answers along accountable_to to the nearest HOAC above it, and the seat is that HOAC's, a human position (Standard §4.1, §4.2, item 6). Who holds it is never written here: it is the organisation's own mechanism, outside the system (DR-0049).

Position (standard title)	Purpose	Reports to	Run today by
			an AI agent / a human position

"Run today by" records only whether an AI agent or a human position runs it; who holds a human position is the organisation's own mechanism, outside the system and outside the Blueprint. A deputy of a human position is the organisation's own too (DR-0049). Two positions are two parties, and the Blueprint never counts or compares the persons behind positions (Standard §1.1, item 10).

Each position's role pack (job description, responsibilities, limits, permissions, checklists, KPIs, coverage, handover, first week) is carried in the role cards of the capabilities it performs.



Section 04

Standard capabilities

Every AIOC has these: Chief of Staff, Input socket and Panel, and an Office where a position needs more than one Chief of Staff. Their **default definitions** are in docs/STANDARD-CAPABILITIES.md (49GAION level). Write "default" in a part to adopt the default text unchanged, or write this AIOC's own text and list the difference in §11.

4.1 Office

Definition

<default | this AIOC's text> · form: <single Chief of Staff | Office of several Chiefs of Staff>

Process (SOP)

<default | this AIOC's text>

Role card

<one card per Chief of Staff in the Office, Annex A.3 fields>

Fixed test

<test ID and version> · result NOT_RUN

4.2 Chief of Staff

Definition

<default | this AIOC's text> · it is this AIOC's voice to its HOAC seat; input arrives only through the typed ports and the Input socket

Process (SOP)

<default | this AIOC's text>

Role card

<Annex A.3 fields>

Fixed test

<test ID and version> · result NOT_RUN

4.3 Input socket

Definition

<default | this AIOC's text> · this AIOC's standard format: <format>

Process (SOP)

<default | this AIOC's text> · converters: <one per input type, each with its test>

Role card

<Annex A.3 fields>

Fixed test

<one test per converter + the shared mechanism test> · result NOT_RUN

4.4 Panel

Definition

<default | this AIOC's text>

Process (SOP)

<default | this AIOC's text>

Role card

<Annex A.3 fields>

Fixed test

<test ID and version> · result NOT_RUN



Section 05

Functional capabilities

One block per function the AIOC performs. Use the **four-part capability form** (Annex A). Name each capability by a plain English name, with no code.

The stages, for an AIOC whose work runs in stages. Before the capabilities, a table 5.0 The stages: input, output, done-test, one row per stage: its ID, plain name, run state, gate, exact input, exact output, done-test and the human decision it needs. Each stage has one input, one output and one done-test, and each capability's **Done when** row (Annex A.1) names the gate it closes. A design stage of the Library (the Protocol section The design chain) also states there its release notice to the next stage and how it carries each return to an end.

Sources: DR-0019.

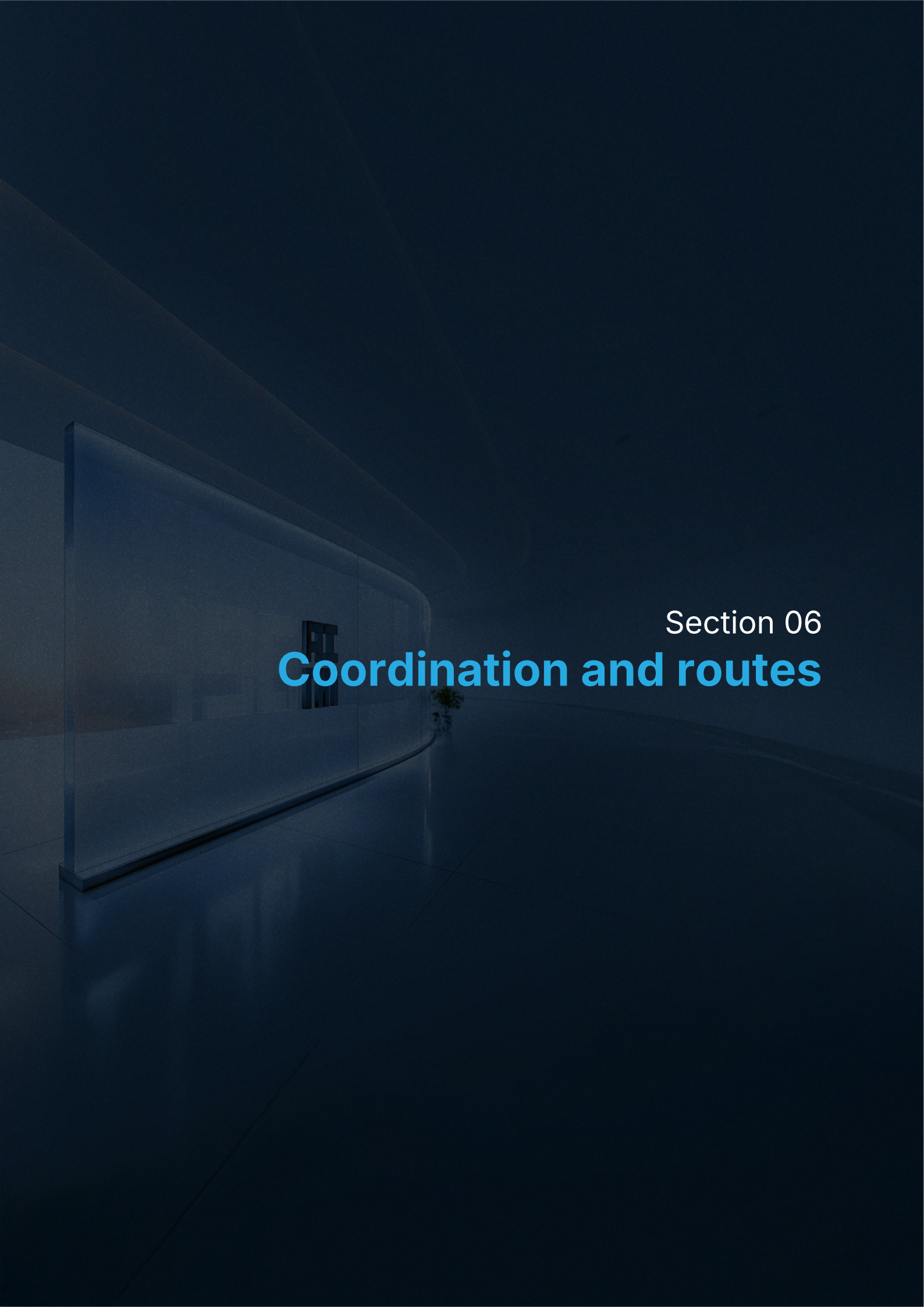
5.N <Capability name>

Definition

Process (SOP)

Role card

Fixed test



Section 06

Coordination and routes

- **Whom it works with**, and the route to each: always through the 49GAION Operating Protocol (one Work Item per hand-off; this AIOC's Chief of Staff as its door).
- **Routing rows this AIOC contributes** to the routing table of its [ORG]AION.
- **Requests and objections.** Owners do not design or change this AIOC's AIs: every proposal, correction or critique is a design-proposal Work Item to **OCO**; where they disagree, the [ORG]HOAC-ACS seat may hold the objection open until it reaches the HOAC seat accountable for OCO, which decides.



Section 07

Records and where things live

Record	Home	Source rule
Writing, decisions, registers	its writing repository <org>aioc-<abbr> (+ -records)	one home for each thing
Code that runs	the code platform named in the [ORG]AION, if any	one home for each thing
Work Items	the shared queue	49GAION Operating Protocol §2



Section 08

Tools and the shared AI gateway

Tool	Existing shared tool, or new	If new: asked of the builder as

All model calls go through the **shared AI gateway**: the tool gateway in the [ORG]AION, with keys, cost and logs in one place; 49GAION runs no gateway for an [ORG]AION (Standard §4.4, item 11).

Connectors and metering. Each tool is reached through a **swappable connector**, recorded by tool **role**; access is granted and withdrawn in the Panel. This AIOC's AI usage and cost are **metered separately** on the shared AI gateway; its **budget cap** is set in §15. A new tool enters only by a sourced proposal, a side-by-side test against the current tool and an update of the part's card.

Tool role	Connector	Access managed in	Part card version
		Panel	

Mechanism register (GOV, 14 October 2026). Every mechanism this AIOC uses, one row each. A row is never deleted, and every change that builds a mechanism cites its row (Standard §3.2).

ID	Mechanism	What it does	Where it lives	Today	Target
				<in force / live / paused / decided, not built>	<keep / build / improve / replace / withdrawn>

Reserve (GOV, 14 October 2026). Ideas not adopted, with the reason and a reconsider-when trigger (Standard §2.1).

ID	Idea	Why held	Reconsider when



Section 09

Fixed tests: index, triggers and scores

Capability	Test ID and version	Evidence level	Last run	Result
Office			—	NOT_RUN
Chief of Staff			—	NOT_RUN
Input socket			—	NOT_RUN
Panel			—	NOT_RUN

- **Re-run triggers:** any change to a model, instructions, this AIOC's card, this blueprint or the 49GAION Operating Protocol re-runs the affected tests automatically; also any change to tools or access, an Input socket converter or the AIOC's standard format, or any shared panel part.
- **Evidence levels:** MEASURED_IN_BUILD_ENVIRONMENT ranks above OWNER_ATTESTED; only a measured score counts toward widening authority (the owner requests with the score attached; the [ORG]HOAC-ACS seat decides and OCO prepares, checks and carries out).
- **Complete** only when every capability's test has a visible result.
- **The Library's cases.** Where the AIOC is made from a Library part, its fixed test keeps the Library part's cases, and where the Library numbers added cases in its own series, this section maps each Library case to the AIOC's own cases that cover it. A Library case with no AIOC case fails the joining test.

Checks that are not fixed tests (GOV, 14 October 2026). Each such check declares how it is placed and when it may block (Standard §3.2). A fixed test and a Human Floor negative test never appear in this table.

Check	Placement (in the producer's loop / on the verdict record / router / after completion)	Budget per item	Degradation order	Blocks only for

- **Discriminating evidence.** A passing check counts only if it has been shown able to fail on the case it guards, and evidence is written by the runner, never by the worker.

Sources: DR-0019.



Section 10

Build: what is generated from this file

Built thing	Generated from	Recorded where
Panel profile	§0, §1, §3, §4.4, §5 role cards	the panel profile and the Panel's release record name this file's commit
Back office	§4.2 role card and SOP (instruction); §4–§5 definitions and SOPs (knowledge)	the back office records the commit it was built or installed from
Card in the protocol AIOC's map	§1	The protocol AIOC's register links this file
Panel Health section	§9 test results; stuck, returned and late Work Items	the Panel

Delivery path: definition change → tests → Panel and Back office update → **one-click rollback** to the previous commit. Both the Panel and the Back office can be rebuilt from any commit of this file.



Section 11

Exceptions

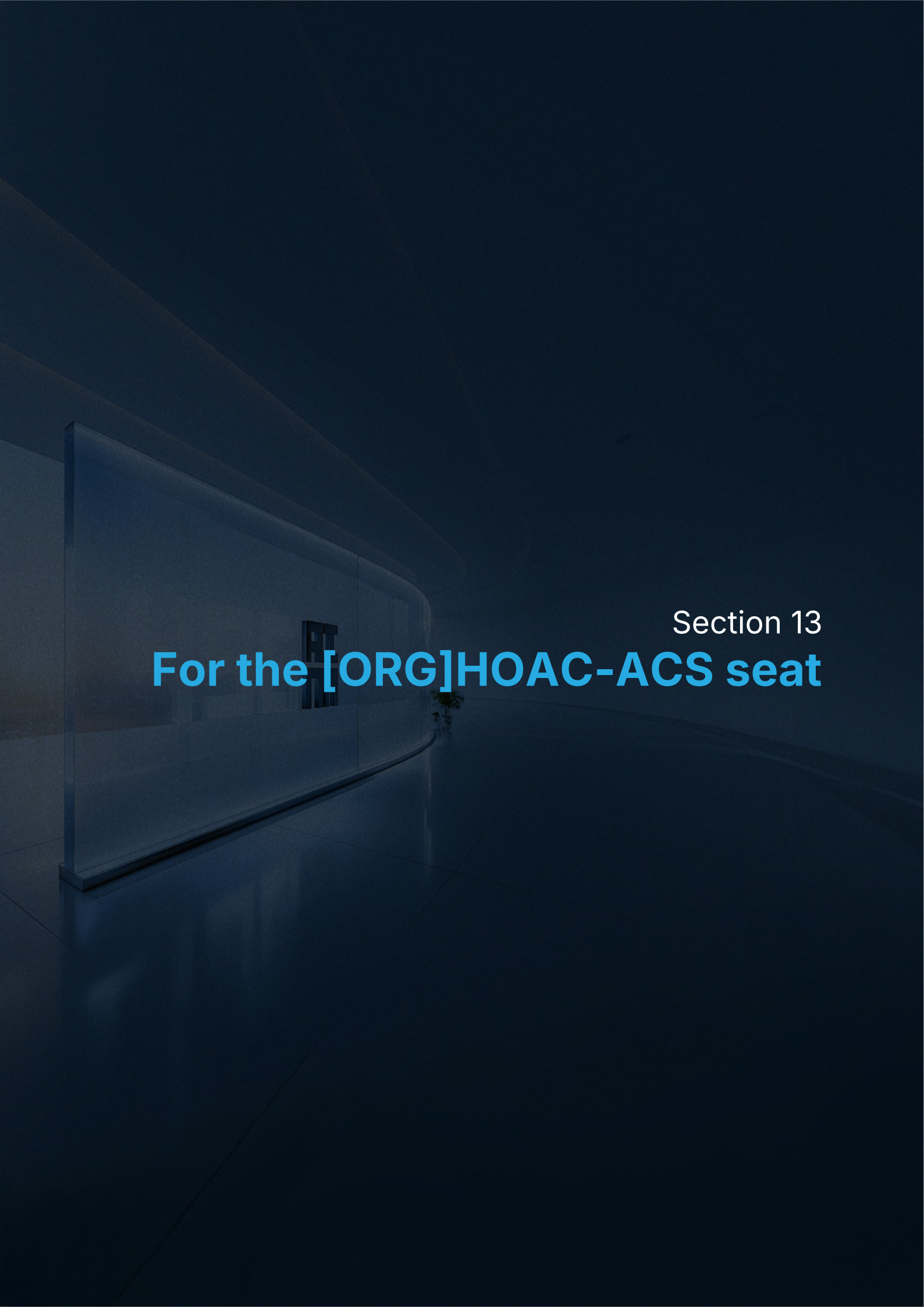
Anything in which this AIOC differs from the template defaults, and why.



Section 12

Lessons and proposals to the mother

One dated line per lesson, each with the test question it added and whether it was proposed to the mother (written without the organisation's details).



Section 13

For the [ORG]HOAC-ACS seat

What the [ORG]HOAC-ACS seat is asked to decide. Money, people and legal matters are listed here so they are seen.



Section 14

Change log

Blueprint version	Date	Change	Template version	Commit



Section 15

Configuration

49GAION fixes the path; who holds each role is configured here (docs/WHAT-IS-AN-AIOC.md §5). Every slot is filled by this organisation or unit; **an AIOC with any empty slot does not pass the joining test** (Protocol §11). Each value records who set it, when, and in which version; a changed value records what it replaced.

Slot	Value	Default offered by 49GAION	Set by	Date	Version	Replaces
HOAC seat accountable for OCO (decides objections held open; Protocol §4.7)		none: must be set				
OCO of this [ORG]AION (receives [ORG]HOAC-ACS requests and carries out approved blueprint changes)		none: must be set				
Discussion limits (how far a discussion may go, and how requests are weighed)		none: must be set				
Waiting deadlines (per status)		none: must be set				
Bounce limit		none: must be set				
Chain depth (maximum depth of Work Item chains; set by the organisation, no fixed cap)		none: must be set				
Halt-request priority		immediate				
Budget cap (AI spend through the shared AI gateway, per AIOC)		none: must be set				
Unattended action (unattended_action: what the Chief of Staff may do after an automatic wake-up with no human present; Protocol §7)		read-record-triage				
Process record defaults (Standard §2.3; spec/schemas/process-record.schema.json in 49g-holding/49GAION-protocol): golden set id, margin, rounds, dwell time, judge, assurance unit, approving seat		none: must be set				

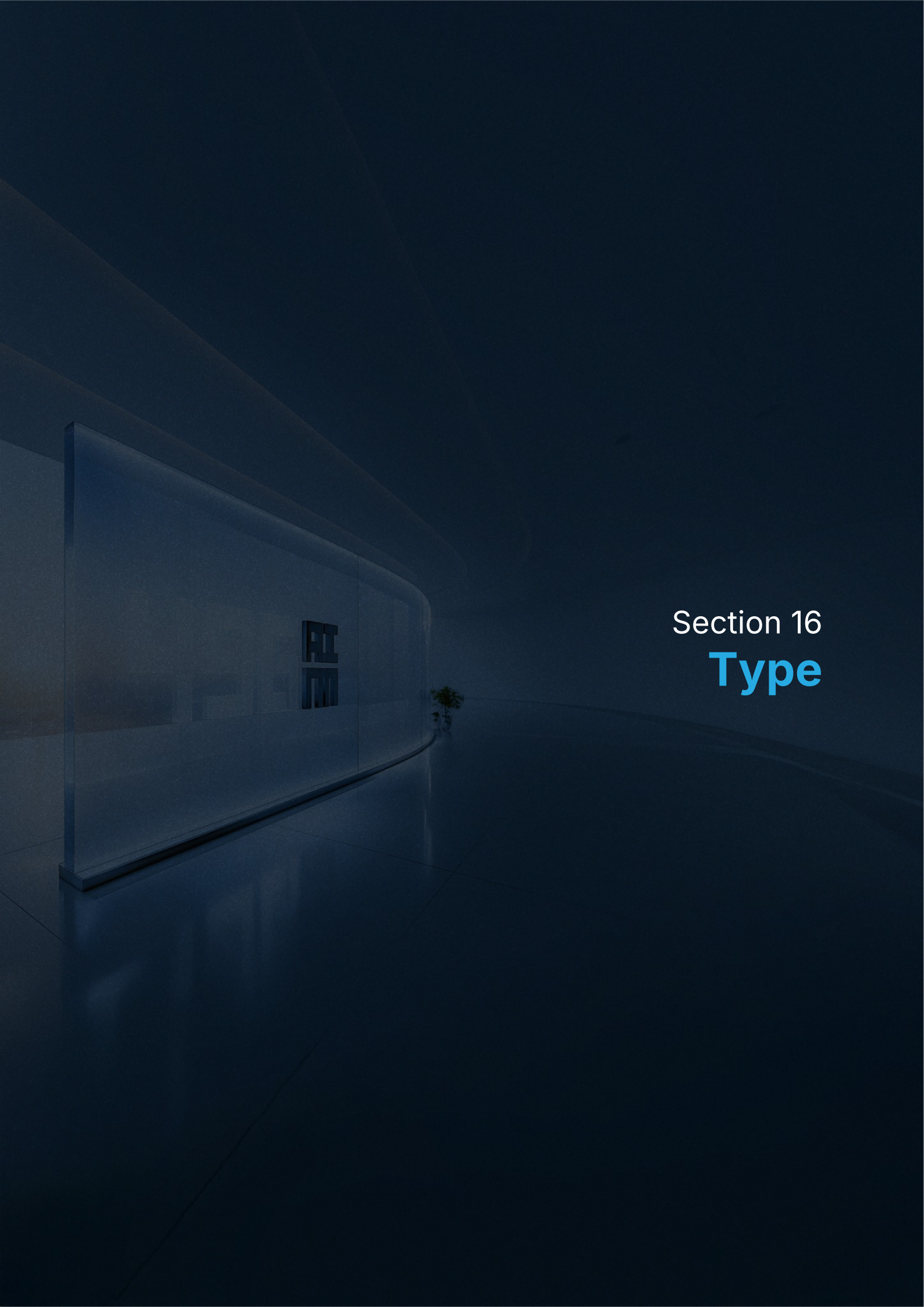
Slot	Value	Default offered by 49GAION	Set by	Date	Version	Replaces
Final authority (GOV; each [ORG]AION fills it; on the objection path, and only there, it is the HOAC seat accountable for OCO, first row, which decides)		none: must be set				

Configuration added (GOV, 14 October 2026). Each value records who set it, when, and in which version. Where 49GAION offers a default, the slot may keep it.

Slot	Value	Default offered by 49GAION	Set by	Date	Version	Replaces
Sensitivity classes of this AIOC's work and the default filler kind of each (Standard §1.5)		none: must be set				
Move and return thresholds per Cell, and their window		none: must be set				
Delegation ladder of each Chief of Staff (decides alone / decides and reports / only prepares / never), per class of act		only prepares, for every class				
Move-on catalogue and stub age cap		stub, default, mock, next item; 5 working days				
Stuck-item thresholds per status, and the repair menu		none: must be set				
Loop breaker (runs of one Cell on one item, per window)		3 runs in 2 hours				
Dispute exit: rounds before the exit, and the classes that go to the human		2 rounds; none: classes must be set				
Stale work: finish window, drop age, cap on work in flight		none: must be set				
Planned halt: grace time and resume order		10 minutes; order must be set				
Waiting time for a personal signature before its hold is raised (Protocol §4.6, item 7)		15 minutes				

Slot	Value	Default offered by 49GAION	Set by	Date	Version	Replaces
Seat load weights and the load line (load and coverage per position only; Standard §4.1, item 13); days before a critical seat is raised again		2 for a seat answering for a unit, 1 for a signing or on-call seat, 0.5 otherwise; line 4; 7 days				
Spend: soft-hold fraction, protected classes of work, degradation order		80%; none: classes must be set				
Capability tiers, tier floor per sensitivity class, engine re-score schedule		none: must be set				
Cadence of each unattended AI Cell		none: must be set				
Duty deputy: ruling life and weekly sample size		90 days; 10				
Quiet windows: share of lower-stakes decisions		none: must be set				

A load line on the people behind positions, if any, is the organisation's own mechanism and record, outside the system, and is never set in a Blueprint (Standard §1.1, item 10; DR-0049).

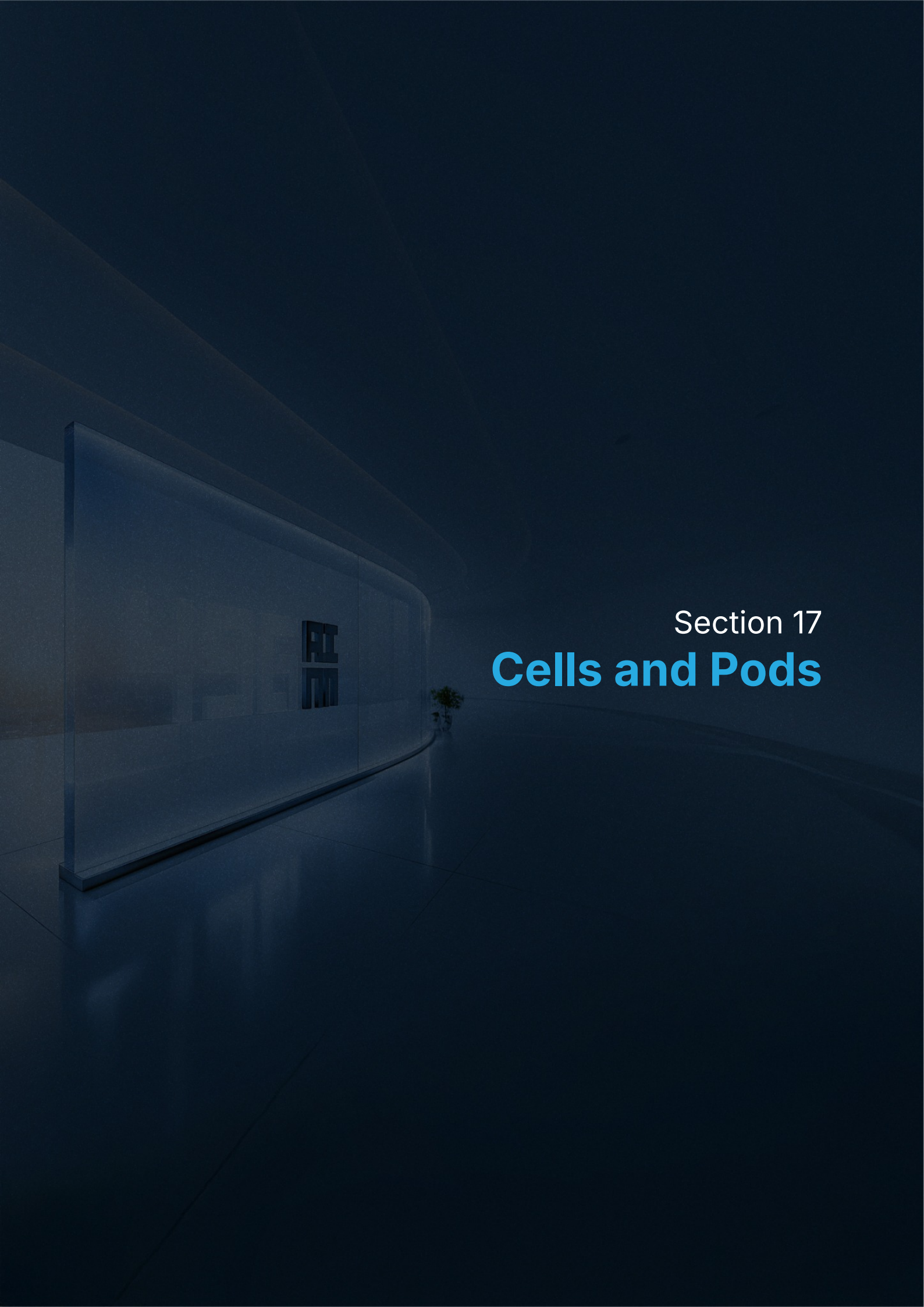


Section 16

Type

- **Type:** <Foundational / Structural / Specialised / Connector / other>, as in §0.
- **Why this AIOC exists in this organisation:** one sentence (the [ORG]AION decides which AIOCs the organisation has).
- **Type-specific test:** <test ID and version> · result NOT_RUN.

The Type sets no count limit, no default connections and no mandatory sections. The Panel is a capability (§4.4), not a type.



Section 17

Cells and Pods

The parts inside this AIOC (docs/CELLS-AND-PODS.md). One card per Cell and per Pod, following spec/schemas/unit-card.schema.json in 49g-holding/49GAION-protocol. The role and nature of this AIOC are read from the type of its top Cell.

Code	Kind	Type	Filler kind	HOAC label	Parent	Attached to	Fixed test (owner)	External ports
<[ORG]CELL-INITIALS>	CELL	<from the organisation's list>	<human / ai / human-or-ai>	<none / [ORG]HOAC-...>	<code>	<the served position's Cell, for an AI assistant>	<test ID and version (seat)>	<none / service and direction>
<[ORG]POD-INITIALS>	POD	<from the organisation's list>	—	<none / [ORG]HOAC-...>	<code>	—	<test ID and version (seat)>	—

Position codes only (GOV, 15 October 2026). Every field of a card that points to a human (accountable_to, label_holder, attached_to, a signer's seat, a fixed test's seat) takes a position code only, in the form <[ORG]CELL-INITIALS>, <[ORG]POD-INITIALS> or <[ORG]AIOC-INITIALS>, never a person's name, initials of a person or internal person ID. Who holds a position is the organisation's own mechanism, outside the system (Standard §5.2, Position Principle; DR-0049).

Complete card forms. The blocks below carry every key the unit card schema requires and are the ones a Blueprint fills. A HOAC label sits only on a Cell whose filler kind is human (Standard §4.1, item 11). An attached AI Cell sits in this Blueprint, has this AIOC (or its Pod) as parent, and names the served position's Cell in attached_to.

```
cell:
  unit_kind: "CELL"
  code: "<[ORG]CELL-INITIALS>"
  full_name: "<Full name>"
  parent: "<code of this AIOC or of its Pod>"
  accountable_to: "<code of the human Cell carrying a HOAC label that this Cell answers to>"
  # hoac: "<[ORG]HOAC-ACS | [ORG]HOAC-TM | [ORG]HOAC-GOV>"   # only on a Cell whose filler_kind is human; omit if none
```

```
  # authority_scope: { decision_types: ["<listed type>"], expires: "<YYYY-MM-DD>",
authority: "<constituted | delegated>" }   # every Cell carrying a HOAC label (Stan
dard §4.4, item 8)

  # attached_to: "<code of the served position's Cell>"   # only for an attached AI
 Cell; omit otherwise

  type: "<from the organisation's list>"

  filler_kind: "<human | ai | human-or-ai>"

  card_version: "<vN.N.N>"

  version: "<N.N.N>"

  lineage: { id: "<code it derives from>", version: "<N.N.N>", relation: "<original
 | derived | merged | split>" }

  lifecycle_state: "<proposed | designed | built | joined | live | replaced | retir
ed>"

  fixed_test: { id: "<test ID>", version: "<vN>", owner: "<code of the human Cell t
hat owns the test>" }

  ports:                                  # one input and one output of each kind,
plus exactly one stop door (Standard §6.1)

    - { unit: "<code>", port: "ai-in", direction: "input", type: "AI", identificati
on: "technical-key", accepted_content: ["<content>"], authority_limit: "<limit>", a
udit_trail: "<where recorded>", required_tests: ["<test ID and version>"], enforcem
ent: "shared-gateway-outside-unit" }

    - { unit: "<code>", port: "ai-out", direction: "output", type: "AI", identifica
tion: "technical-key", accepted_content: ["<content>"], authority_limit: "<limit>",
 audit_trail: "<where recorded>", required_tests: ["<test ID and version>"], enforc
ement: "shared-gateway-outside-unit" }

    - { unit: "<code>", port: "hu-in", direction: "input", type: "HU", identificati
on: "live-human-sign-in-at-gateway", accepted_content: ["<content>"], authority_lim
it: "<limit>", audit_trail: "<where recorded>", required_tests: ["<test ID and vers
ion>"], enforcement: "shared-gateway-outside-unit" }

    - { unit: "<code>", port: "hu-out", direction: "output", type: "HU", identifica
tion: "live-human-sign-in-at-gateway", accepted_content: ["<content>"], authority_l
imit: "<limit>", audit_trail: "<where recorded>", required_tests: ["<test ID and ve
rsion>"], enforcement: "shared-gateway-outside-unit" }

    - { unit: "<code>", port: "stop", direction: "input", type: "HU", identificatio
n: "live-human-sign-in-at-gateway", accepted_content: ["stop"], authority_limit: "s
top only", audit_trail: "<where recorded>", required_tests: ["<test ID and version>
"], enforcement: "shared-gateway-outside-unit", stop_door: true }
```

A Pod example (a Pod holds Cells only; a Pod with a HOAC label names its label_holder; a board behind a position is the organisation's own, DR-0049):

```
pod:
```

```
  unit_kind: "POD"

  code: "<[ORG]POD-INITIALS>"

  full_name: "<Full name>"

  parent: "<code of this AIOC>"

  accountable_to: "<code of the human Cell carrying a HOAC label that this Pod answ
ers to>"

  type: "<from the organisation's list>"

  inner_units: ["<[ORG]CELL-INITIALS>"]

  hoac: "<[ORG]HOAC-ACS | [ORG]HOAC-TM | [ORG]HOAC-GOV>"   # omit if none

  label_holder: "<code of the human Cell, in this Pod, that holds the label>"

  card_version: "<vN.N.N>"

  version: "<N.N.N>"

  lineage: { id: "<code it derives from>", version: "<N.N.N>", relation: "<original
 | derived | merged | split>" }

  lifecycle_state: "<proposed | designed | built | joined | live | replaced | retir
ed>"

  fixed_test: { id: "<test ID>", version: "<vN>", owner: "<code of the human Cell t
hat owns the test>" }

  ports:                                  # one input and one output of each kind,
plus exactly one stop door (Standard §6.1)

    - { unit: "<[ORG]POD-INITIALS>", port: "ai-in", direction: "input", type: "AI",
 identification: "technical-key", accepted_content: ["<content>"], authority_limit:
 "<limit>", audit_trail: "<where recorded>", required_tests: ["<test ID and version
>"], enforcement: "shared-gateway-outside-unit" }

    - { unit: "<[ORG]POD-INITIALS>", port: "ai-out", direction: "output", type: "AI
", identification: "technical-key", accepted_content: ["<content>"], authority_limi
t: "<limit>", audit_trail: "<where recorded>", required_tests: ["<test ID and versi
on>"], enforcement: "shared-gateway-outside-unit" }

    - { unit: "<[ORG]POD-INITIALS>", port: "hu-in", direction: "input", type: "HU",
 identification: "live-human-sign-in-at-gateway", accepted_content: ["<content>"],
authority_limit: "<limit>", audit_trail: "<where recorded>", required_tests: ["<tes
t ID and version>"], enforcement: "shared-gateway-outside-unit" }

    - { unit: "<[ORG]POD-INITIALS>", port: "hu-out", direction: "output", type: "HU
", identification: "live-human-sign-in-at-gateway", accepted_content: ["<content>"]
, authority_limit: "<limit>", audit_trail: "<where recorded>", required_tests: ["<t
est ID and version>"], enforcement: "shared-gateway-outside-unit" }

    - { unit: "<[ORG]POD-INITIALS>", port: "stop", direction: "input", type: "HU",
identification: "live-human-sign-in-at-gateway", accepted_content: ["stop"], author
```

```
ity_limit: "stop only", audit_trail: "<where recorded>", required_tests: ["<test ID
 and version>"], enforcement: "shared-gateway-outside-unit", stop_door: true }
```

- A Pod holds Cells only; anything that needs nesting is an AIOC.
- An external service, API or tool is a port on the card of the Cell that receives or sends, never a Cell; an external AI agent acting directly in the work is an AI Cell.



Annex A

The four-part capability form (normative for every capability)

A.1 Definition — must contain

Field	Content
Name	plain English, no code
What it is	one paragraph
Purpose	one sentence
Scope	does / does not
Inputs	what, from whom, in what format
Outputs	what, to whom, in what format
Limits	what it may never do
Sources	the governing texts it rests on, by ID and version
Status	draft, in force or replaced
Done when	the test or gate that closes it, by name: what a reader can check to know the capability's work is done. Required for every functional capability; a standard capability takes it from its fixed test.

Sources: DR-0019.

A.2 Process (SOP) — must contain

Field	Content
Trigger	what starts it
Steps	numbered; each: position — action — output — evidence
States	the states an item passes through and who moves it
Failure and recovery	raise, never guess; two failures on the same step stop and go to the owner
Records	what it writes, and where
Cadence	daily / weekly / monthly / quarterly, if any
Measures	name — definition — target (no target invented)

A.3 Role card — must contain

Field	Content
Name	plain English
Position and seat	the position; its accountable [ORG]HOAC-ACS seat
Performed by	a bot, an agent or a human position; who holds a human position is the organisation's own mechanism, outside the system
Purpose	one line
Authority limits	alone / needs the owner / never — mirroring the panel switch
Write location	the single place it writes
Reports to	normally the Chief of Staff; the owner directly only in a board session
Escalation	where and how
Tools and access	which, through the panel; never credentials in instructions
Cover	for an AI role, what takes over when it cannot; for a human position, nothing: its cover is the organisation's own mechanism
Fixed test and score	the test ID
Card version and status	—

A role card never counts or compares the persons behind positions: two positions are two parties, even when one natural person holds both (Standard §1.1, item 10).

A.4 Fixed test — must contain

Field	Content
Test ID and version	frozen once used; items are struck by adjudication, never renumbered
Cases	each with pass criteria, written before use
Evidence level	MEASURED or OWNER_ATTESTED
Re-run triggers	§9
Lessons added	each lesson adds at least one case
Where results are kept	§9 and the panel
Last result	NOT_RUN until run; never reported as PASS

A.5 Bot identity card — must contain

Every bot performing a role card has one identity card.

Field	Content
Role	the role card it performs (A.3)
Account	its **own separate account**; never a human's account and never shared with another bot
Allowed actions	what it may do alone

Field	Content
Ask first	the actions that require a **prior request to the accountable human**, and how the request is made
Card version and status	—

Schema: 49GAION-protocol spec/schemas/bot-identity-card.schema.json.

49GAION AIOC Blueprint Template v2.1.1 · Powered by 49GAION


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