

# Task Manager

**Doc type:** technical · **Version:** v0.1 · **Status:** published · **Module slug:** task-manager  
**Exported:** 2026-05-15 11:10 UTC · **By:** anonymous

## Task Manager – Technical Specification

### 1. Module Purpose & Scope (Authoritative)

---

Task Manager is the calendar-backed system of record for operational work across Primoro. It ensures that all planned and unplanned work — including tasks, callbacks, checklists, and work packages — is created as time-bound, visible work; explicitly owned by a role or individual; aligned to real staff availability; completed within defined time windows or SLAs; and fully auditable. Task Manager replaces private to-do lists, inbox chasing, and informal messaging with one governed work system: if work matters, it exists in the calendar.

It governs:

- Creating, assigning, progressing, and completing tasks, callbacks, checklists, and work packages
- Role-first assignment, dynamic claiming, and SLA tracking across all work items
- Template-based and recurrence-driven generation of task and work-package instances
- Kanban board, list, and queue views with audit-logged state transitions
- Checklist execution with evidence capture (notes, comments, photos, and file attachments)
- Archival and automated retention for completed work and evidence

It explicitly does not:

- Define staff availability or shift structure (owned by Rota Manager)
- Deliver or store messages and conversations (owned by Communication Hub)
- Own appointment booking logic or Reschedule Job tracking (owned by Appointment Manager)
- Store clinical notes, HR records, or payment data
- Execute autonomous actions without explicit human confirmation

### 2. Ownership & Responsibilities

---

#### 2.1 Task Manager IS Responsible For

- End-to-end lifecycle management of tasks, callbacks, checklists, and work packages from creation through archival
- Maintaining an immutable, inspection-ready audit trail of all task and checklist events
- Exposing integration surfaces consumed by Communication Hub, Rota Manager, Appointment Manager, Aftercare Manager, Aiden, AI Meeting Notes, AI Guardian, AI Quality Monitor, AI Concierge, Lab Manager, Knowledge Training & Learning, HR & People Manager, Inventory & Compliance Manager, Referral Manager, Recall & Reconnect, Financial Insights, Product Shop, Digital Forms, Smart Dashboards, and Admin Control Plane

- Computing and owning the `Overdue` boolean flag; consuming surfaces MUST NOT compute it independently
- Reverting in-flight task ownership to the associated role queue when Access Manager emits an access-revocation event
- Governing execution lifecycle for tasks generated by all external event sources, regardless of which module originated the task

## 2.2 Task Manager IS NOT Responsible For

- Staff availability definition — owned by Rota Manager
- Messaging delivery or conversation storage — owned by Communication Hub
- Appointment booking logic and Reschedule Job ownership — owned by Appointment Manager
- Compliance task state definition (Scheduled / Due / Overdue / Escalated) — owned by Inventory & Compliance Manager; Task Manager maps these states onto its own lifecycle
- Learning pathway content and assignment logic — owned by Knowledge, Training & Learning
- HR workflow policy rules and sequence logic — owned by HR & People Manager
- Tenant health-event detection and task-creation decisions — owned by Admin Control Plane
- Task execution for aftercare workflows — task execution is governed exclusively by Task Manager; Aftercare Manager owns aftercare record management and delegates task execution here
- Clinical records, payments, or HR personnel records

## 3. Core Objects (Normative)

---

### 3.1 Task (Canonical Artefact)

A Task is a governed digital artefact representing a discrete unit of operational work. Every Task MUST:

- exist as a time-bound calendar event
- have a defined purpose and explicit ownership
- have a lifecycle state at all times
- retain completion evidence

Minimum required fields:

- `TaskId` (immutable UUID)
- `Title`, `Description`, `Category`
- `DueDateTime` OR `TimeWindowStart` / `TimeWindowEnd`
- `Origin` — **one of**: `Manual`, `Callback`, `Workflow`, `Template-Generated`, `AI-Suggested`
- `OwnerType` (`User` or `Role`), `OwnerId`
- `Status` (see §3.2)
- `SLAState` (`On-track`, `At-risk`, `Breached`)
- `Priority`
- `Confidential` (boolean)

- `LinkedEntities` — back-links to originating records (e.g. `MeetingNoteRecord`, lab case reference, HR workflow instance, compliance record, Reschedule Job reference)
- `Overdue` (boolean; computed and owned by Task Manager)
- `CompletedTimestamp`, `CompletedByUserId`
- `ArchivedAt`, `RetentionPolicyId`
- `AuditLog` (immutable)

### 3.2 Task State Machine (Authoritative)

States:

- `Created`
- `Active`
- `In Progress`
- `Blocked` (optional)
- `Completed`
- `Cancelled` (with mandatory reason)
- `Archived` (system state; post-completion, applied by retention policy)

Rules:

- All state transitions are explicit, time-stamped, and immutable in the audit log.
- A task cannot return to `Created` once it has reached `Active` or beyond.
- `Completed` and `Cancelled` tasks may not transition to any operational state; they proceed only to `Archived`.
- `Archived` is a terminal system state; it cannot be reversed.
- Cancellation MUST record a reason.
- Kanban board drag-and-drop constitutes an explicit state transition and MUST be audit-logged identically to programmatic transitions.
- Only authorised users (per Access Manager RBAC) may trigger specific transitions; see §9.
- The seven states above are canonical and exhaustive. Consuming surfaces (including the Surgery & Decon Day View tablet rendering layer) MUST NOT introduce or imply additional states. Each state maps to a defined visual treatment in consuming surfaces; any mismatch MUST be treated as an integration defect and raised against the Task Manager module owner.

### 3.3 Checklist / Checklist Item (Canonical Artefact)

Checklists may belong to a Task or a Work Package. Minimum required fields:

- `ChecklistId` (immutable UUID)
- `ParentId` — FK to owning Task or Work Package
- `Items[]` — each with an immutable `ChecklistItemId`, `CompletionState`, and optional evidence payload (notes, comments, photos, attachments)
- `AuditLog` (immutable)

Checklists provide execution evidence; they do not replace or alter the owning Task's lifecycle state.

### 3.4 Work Package (Canonical Artefact)

A Work Package is a governed collection of related Tasks (each optionally with checklists), delivered as a single operational unit (e.g. Daily Opening, Weekly Decon).

Minimum required fields:

- `WorkPackageId` (immutable UUID)
- `Title, Description`
- `TimeWindowStart / TimeWindowEnd`
- `OwnerType` (User Or Role), `OwnerId`
- `Status` (mirrors Task lifecycle states)
- `Tasks[]` — ordered list of constituent Task references
- `LinkedEntities`
- `AuditLog` (immutable)

Requirements:

- Calendar-backed and time-bound
- One or more governed tasks per package
- Evidence captured at task or checklist-item level
- Role-first assignment and claiming (configurable per package)
- Fully auditable across creation, execution, completion, and archival

### 3.5 Templates & Recurrence

Templates enable consistent operational delivery.

Task Templates define: title, description, category, default owner (user or role), due-window rules, checklist structure, and evidence requirements.

Work Package Templates define: tasks within the package, checklist structures, assignment rules, and timing rules.

Recurrence rules recreate new instances on a defined schedule (daily / weekly / monthly / custom). Each generated instance MUST:

- carry a fresh, independent audit trail
- link back to its originating template via `LinkedEntities`
- be generated idempotently (no silent duplicates)

## 4. Creation & Update Sources

---

### 4.1 Manual Creation

Authorised staff may create ad-hoc tasks and checklists. Permissions govern assignment scope, evidence requirements, and cancellation rights.

### 4.2 Callbacks (CORE)

Patient callbacks are a CORE capability. Callbacks are tasks with due windows, explicit ownership, and full audit logging. Callbacks and follow-up tasks may generate one another.

## 4.3 Event-Driven Tasks

Tasks are generated from the following sources:

- Appointments (see Appointment Manager integration below)
- Communication Hub conversations and message-to-work conversions
- Workflow completion events
- Aftercare Manager triggers
- Inventory & Compliance Manager compliance cycles
- Exception sources (e.g. Aiden quality-monitor alerts, AI Quality Monitor findings)
- AI Meeting Notes (confirmed action items and retention review tasks)
- AI Guardian (finding follow-up and actionable alert routing)
- AI Concierge (unresolved call follow-ups and recovery tasks)
- Lab Manager SLA breach escalations
- Knowledge, Training & Learning pathway assignments
- HR & People Manager policy-driven workflow steps and Critical Risk escalations
- Referral Manager SLA breach escalations and referral-state follow-up actions
- Recall & Reconnect non-clinical personal follow-up work items
- Financial Insights anomaly and risk follow-up actions
- Product Shop in-clinic fulfilment and exception-handling actions
- Digital Forms form-review and mandatory re-capture actions
- Rota Manager absence-change events (where absence triggers downstream work, e.g. rescheduling callbacks)
- Admin Control Plane / Tenant Health Monitor health events

All event-driven tasks MUST carry `Origin: Workflow` (or `Origin: AI-Suggested` for Aiden-confirmed tasks) and MUST include a `LinkedEntities` back-link to the originating record in the source module.

### Appointment Manager — Appointment Lifecycle and Reschedule Follow-Up

Appointment Manager emits two categories of event that may result in task creation in Task Manager:

- **Lifecycle transition tasks.** When an appointment transitions to a state that creates a downstream task obligation (e.g. completed, no-show, or cancelled), Appointment Manager MUST emit an `AppointmentTaskTriggerEvent` to Task Manager. Task Manager MUST create the corresponding task with `Origin: Workflow` and a `LinkedEntities` back-link to the originating appointment record. The triggering conditions and task payload are defined in the Appointment Manager integration contract.
- **Reschedule follow-up tasks.** When practitioner absence impacts booked appointments, Appointment Manager creates and owns a Reschedule Job, tracking each affected patient independently. Reschedule Jobs are owned and managed by Appointment Manager, not Task Manager. Where Appointment Manager emits an event requiring operational follow-up by practice staff (e.g. a patient contact action), that follow-up work MAY be created as an event-driven task in Task Manager with `Origin: Workflow` and a `LinkedEntities` back-link to the relevant Reschedule Job reference.

Task Manager does not duplicate Appointment Manager's booking logic.

## AI Meeting Notes — Confirmed Action Tasks and Retention Review Tasks

AI Meeting Notes is a recognised event-driven source for two categories of task:

- **Confirmed action tasks.** When a staff member confirms an `ActionCandidate` in AI Meeting Notes, that module MUST create a task in Task Manager. The task MUST carry `Origin: Workflow`, a `LinkedEntities` back-link to the originating `MeetingNoteRecord`, and title, owner, and due date supplied by the confirming user at sign-off time. No task is created autonomously; explicit human confirmation in the AI Meeting Notes sign-off flow is the sole trigger.
- **Retention review tasks.** The AI Meeting Notes sign-off flow MAY create a retention review task with a standardised title pattern, an assigned owner, a defined due date, and a `LinkedEntities` reference to the `MeetingNoteRecord`.

In both cases, Task Manager treats AI Meeting Notes as an external caller; no autonomous creation occurs without an explicit human confirmation action in that module. Task creation is idempotent: re-submission of the same confirmed action MUST NOT produce duplicate tasks.

## AI Guardian — Finding Follow-Up Tasks

AI Guardian identifies quality and safety findings and routes actionable outputs to Task Manager. When AI Guardian determines that a finding requires staff follow-up, it MUST emit an event that results in a task being created in Task Manager. The task MUST carry `Origin: Workflow` and a `LinkedEntities` back-link to the originating finding record in AI Guardian. AI Guardian is responsible for determining when a finding warrants task creation and for defining the default owner (role or named user); Task Manager governs the task's execution lifecycle, SLA tracking, and audit trail thereafter.

## AI Quality Monitor — Quality Finding Remediation Tasks

AI Quality Monitor creates structured remediation artefacts (tasks and alerts) when quality issues are detected. When a quality finding requires staff remediation action, AI Quality Monitor MUST emit an event resulting in a task being created in Task Manager. The task MUST carry `Origin: Workflow` and a `LinkedEntities` back-link to the originating quality finding record. Task Manager assigns the standard lifecycle, SLA tracking, and audit trail from the point of creation; AI Quality Monitor retains responsibility for finding detection and remediation-task initiation.

## AI Concierge — Unresolved Call and Recovery Follow-Up Tasks

When an AI Concierge call cannot be resolved autonomously, the session MUST transition to a `Follow-Up Created` state and generate a task in Task Manager before the session closes. The task MUST carry `Origin: Workflow` and a `LinkedEntities` back-link to the originating call session record. Recovery follow-up tasks (e.g. where a patient requires a callback after a failed automated interaction) follow the same contract. Task Manager governs the follow-up task's lifecycle; AI Concierge is responsible for determining when escalation to a task is warranted.

## Lab Manager — SLA Breach Escalation Tasks

When a lab case breaches its SLA, Lab Manager emits an escalation event and Task Manager MUST create an escalation task. The task payload supplied by Lab Manager MUST include at minimum:

- `Origin: Workflow`
- A `LinkedEntities` reference to the relevant lab case
- Escalation context (reason and breach timestamp)

- A default owner (role or named user as determined by Lab Manager configuration)

Task Manager assigns the standard lifecycle to the escalation task from the point of creation.

### Knowledge, Training & Learning — Pathway Assignment Tasks

When Knowledge, Training & Learning assigns a learning pathway — whether triggered automatically on onboarding or role change, or manually by a manager — it MUST emit an event that results in a task being created in Task Manager. Task Manager is responsible for deadline tracking, reminders, escalation visibility, and completion recording against that task. The task MUST carry `Origin: Workflow` and a `LinkedEntities` back-link to the relevant learning pathway record. Knowledge, Training & Learning defines the learning requirement; Task Manager governs its execution lifecycle.

### HR & People Manager — Policy-Driven Workflow Steps and Critical Risk Escalations

HR & People Manager executes formal policy workflows (including disciplinary and grievance processes) via Task Manager, and MUST also trigger escalation tasks when a leave classification reaches Critical Risk status. Specifically:

- **Policy workflow step tasks.** Each step in an HR workflow requiring staff action MUST generate a task with `Origin: Workflow` and a `LinkedEntities` back-link to the relevant HR workflow instance.
- **Critical Risk escalation tasks.** When HR & People Manager classifies an absence or leave event as Critical Risk, it MUST emit an event that results in a task being created in Task Manager for managerial follow-up, with `Origin: Workflow` and a `LinkedEntities` back-link to the relevant HR record.

HR & People Manager remains responsible for workflow logic, policy rules, and the determination of risk classification; Task Manager is responsible for task creation, ownership, lifecycle progression, and audit trail.

### Inventory & Compliance Manager — Compliance Cycle Tasks

Compliance Tasks have a distinct five-state lifecycle within Inventory & Compliance Manager (Scheduled → Due → Completed → Overdue → Escalated). These states map onto Task Manager's governed lifecycle as follows:

Inventory & Compliance Manager state	Task Manager lifecycle effect
Scheduled	Task exists in <code>Created</code> or <code>Active</code> state
Due	Task moves to <code>Active / In Progress</code>
Completed	Task moves to <code>Completed</code>
Overdue	<code>SLAState</code> set to <code>Breached</code> ; task remains in <code>Active</code> or <code>In Progress</code>
Escalated	Manual escalation visibility surfaced; ownership reassignment is manual

Task Manager MUST accept the compliance task payload from Inventory & Compliance Manager with `Origin: Workflow`, a `LinkedEntities` back-link to the compliance record, and the relevant SLA window. Both modules MUST agree the state-mapping contract during integration design.

### Referral Manager — SLA Breach Escalation and Referral State Follow-Up Tasks

Referral Manager generates referral-linked tasks and escalation signals via Task Manager. Two categories of task may be created:

- **SLA breach escalation tasks.** When a referral breaches its SLA, Referral Manager MUST emit an escalation event resulting in a task being created in Task Manager with `Origin: Workflow` and a `LinkedEntities` back-link to the referral record.
- **Referral state follow-up tasks.** Referral state transitions that require staff action (e.g. triaging an inbound referral, ownership reassignment) MAY generate a follow-up task in Task Manager with `Origin: Workflow` and a `LinkedEntities` back-link to the referral record.

Referral Manager is responsible for determining when a state transition or SLA breach warrants task creation; Task Manager governs execution lifecycle, SLA tracking, and audit trail thereafter.

### Recall & Reconnect — Personal Follow-Up Work Items

When Recall & Reconnect automation cannot progress safely (e.g. due to clinical complexity or patient preference), it MUST generate a non-clinical personal follow-up work item in Task Manager rather than continuing automated outreach. The task MUST carry `Origin: Workflow` and a `LinkedEntities` back-link to the originating recall or reconnect record. Task Manager governs the follow-up task's lifecycle; Recall & Reconnect is responsible for determining when escalation to a human-owned task is warranted.

### Financial Insights — Anomaly and Risk Follow-Up Tasks

When Financial Insights detects a financial anomaly or risk that requires governed follow-up, it MUST emit an event resulting in a task being created in Task Manager. The task MUST carry `Origin: Workflow` and a `LinkedEntities` back-link to the originating Financial Insights record. Financial Insights is responsible for anomaly detection and the determination of when a task is warranted; Task Manager governs task execution lifecycle, SLA, and audit trail.

### Product Shop — In-Clinic Fulfilment and Exception-Handling Tasks

When Product Shop determines that a staff action is required — for example, in-clinic fulfilment of an order or exception handling where automated processing cannot proceed — it MUST emit an event resulting in a task being created in Task Manager. The task MUST carry `Origin: Workflow` and a `LinkedEntities` back-link to the originating Product Shop order or exception record. Product Shop is responsible for determining when human intervention is required; Task Manager governs the task's execution lifecycle.

### Digital Forms — Form Review and Re-Capture Tasks

When Digital Forms identifies that a form requires staff review (e.g. a form has expired or must be re-captured with documented justification), it MAY emit an event resulting in a task being created in Task Manager. The task MUST carry `Origin: Workflow` and a `LinkedEntities` back-link to the originating form record. Digital Forms is responsible for determining when a review or re-capture task is warranted; Task Manager governs the task's execution lifecycle.

### Rota Manager — Absence-Driven Follow-Up Tasks

Rota Manager emits a canonical `AbsenceChangeEvent` on all absence state transitions. Where an absence change creates downstream operational work that cannot be handled automatically (e.g. rescheduling callbacks previously owned by the absent staff member), an event-driven task MAY be created in Task Manager with `Origin: Workflow` and a `LinkedEntities` back-link to the relevant absence record. The determination of when an absence event warrants task creation is governed by the consuming integration configuration; Task Manager does not replicate Rota Manager's absence or availability logic.

### Admin Control Plane / Tenant Health Monitor — Health Event Tasks

The Admin Control Plane Tenant Health Monitor may auto-generate support tasks and submit them to Task Manager. Tasks originating from Admin Control Plane MUST carry `Origin: Workflow` and MUST be scoped to the relevant tenant. Default ownership is assigned to a support or administrator role as configured in Admin Control Plane. Task Manager treats Admin Control Plane-originated tasks identically to other event-driven tasks in terms of lifecycle, SLA, and audit trail. Admin Control Plane is responsible for determining when a health event warrants task creation; Task Manager governs execution thereafter.

## 4.4 Template-Driven Generation

- On-demand creation from templates
- Scheduled recurrence from templates
- Idempotent generation (no silent duplicates)
- Fully auditable creation lineage including template and recurrence linkage

## 4.5 AI-Assisted Suggestions (Where Enabled)

Aiden (Primoro AI Assistant) MAY suggest tasks or highlight risk. Aiden communicates task suggestions via its canonical interface models:

- **AIAction** — carries `ActionType` (e.g. `CreateTask`), `TargetModule` (`task-manager`), and `RequiresConfirmation` (always `true` for task creation).
- **AIIntent** — carries `IntentType`, `ConfidenceScore`, and `SupportedActions[]` listing candidate `AIAction` entries.

When `TargetModule` is `task-manager`, Aiden's `ActionType` MUST be `Suggest` only. Aiden MUST NOT emit `ActionType` values of `Create`, `Assign`, or `Complete` targeting Task Manager; Task Manager MUST reject any `AIAction` payload carrying those action types regardless of other field values. This constraint is non-negotiable and applies at both the Aiden boundary and at Task Manager's inbound validation layer.

Task Manager MUST surface the suggested action to an authorised user for explicit confirmation before any task is created. The `ConfidenceScore` from `AIIntent` MAY be displayed to the user to aid judgement but MUST NOT be used to bypass confirmation. `RequiresConfirmation: true` is non-negotiable; Task Manager MUST reject any `AIAction` payload where this flag is absent or false. Confirmed AI-suggested tasks carry `Origin: AI-Suggested` and retain a `LinkedEntities` back-link to the originating `AIIntent` for audit purposes.

## 5. Delivery Surfaces & Access (Authoritative)

---

### 5.1 Web Portal

The Web Portal provides planning and oversight capability for authorised staff:

- Workload and SLA views
- Planning and exception management
- Template and recurrence administration
- Read-only audit inspection
- Exception view for tasks requiring manual reassignment (e.g. after access revocation with no determinable role queue)

### 5.2 Tablet App (Surgery & Decontamination)

Context-specific execution views optimised for point-of-work completion:

- Checklist-first UX
- Explicit user attribution on shared devices
- Device-level authentication enforced via Access Manager

Work package tasks surfaced on surgery and decontamination tablets MUST expose the following fields to the tablet rendering layer:

Field	Description
<code>DueTime</code>	Scheduled due time within the task's time window
<code>Status</code>	Current lifecycle state
<code>CompletedTimestamp</code>	Date-time of completion (null if not yet completed)
<code>CompletedByUserId</code>	Attributed completing user (null if not yet completed)
<code>Overdue</code>	Boolean; <code>true</code> when current time exceeds <code>DueTime</code> and <code>Status</code> is not <code>Completed</code> or <code>Cancelled</code>

The `Overdue` flag MUST be computed and kept current by Task Manager; consuming surfaces such as the Surgery & Decon Day View MUST NOT compute it independently. The Surgery & Decon Day View MUST render each task state using only the canonical states defined in §3.2; no additional or derived states are introduced by the rendering layer.

### 5.3 Patient Mobile App

Task Manager does not surface directly in the patient mobile app. Patient-facing work (e.g. callback requests) is mediated through Communication Hub and Aftercare Manager, which may generate tasks in Task Manager as an event source.

### 5.4 Staff App Mode

Personal and role-based task management for staff on personal devices:

- Personal task list (user-owned tasks filtered by ownership)
- Role-based queues (tasks assigned to the user's role, scoped by location and tenant)
- Task claiming from role-based queues
- Task acknowledgement (distinct from claiming or completing — see below)
- Checklist execution with evidence capture
- Completion at point of work

Staff App Mode surfaces role-scoped views of tasks from Task Manager. The following filtering rules apply to the Staff App Mode task surface:

- Tasks are filtered by the authenticated user's assigned roles and location.
- Confidential tasks are only surfaced where the user holds an explicitly permitted role.

- Overdue tasks are flagged using the `Overdue` boolean computed by Task Manager; Staff App Mode **MUST NOT** recompute this flag.
- Queue views display tasks in `Created` and `Active` states available for claiming, alongside tasks already owned by the authenticated user in any non-terminal state.

**Acknowledgement Action.** Acknowledgement is a first-class interaction in Staff App Mode. A user **MAY** acknowledge a task to confirm awareness without yet claiming ownership or beginning execution.

Acknowledgement **MUST**:

- be available on tasks in `Created` and `Active` states
- be recorded as an explicit, time-stamped, attributed action in the audit log
- not change the task's lifecycle state or ownership
- be visible to supervisors and role-queue managers in list and board views

Where Staff App Mode surfaces role-based queues, the acknowledge and claim actions **MUST** both be available as distinct controls, consistent with the Staff App Mode module's interaction contract.

## 5.5 Engagement Signals

Task Manager emits SLA state changes, completion events, and escalation events as signals available to staff-facing dashboards and the Web Portal workload view. All signals are derived from audited state transitions and carry timestamps and actor attribution. Smart Dashboards consumes these signals — specifically open task queue state and overdue task counts — on a near real-time basis to populate role-specific task queue widgets (TCO, Manager, Practitioner, and Nurse dashboard views). Task Manager is the authoritative source for all such signals; Smart Dashboards **MUST NOT** derive task queue or overdue state independently.

## 6. Integration Contracts

### 6.1 Inbound (this module consumes from)

From module	What	Contract
Rota Manager	Work Package overlay event (shift reference, role, location, linked work package template ID); AbsenceChangeEvent (where absence triggers downstream task creation)	Async event
Appointment Manager	AppointmentTaskTriggerEvent (lifecycle transition task); Reschedule Job follow-up event	Async event
Communication Hub	Conversation-triggered task event; message-to-work conversion task event	Async event
Aftercare Manager	Aftercare trigger event	Async event

AI Meeting Notes	Confirmed <code>ActionCandidate</code> event; retention review task event	Async event
AI Guardian	Finding follow-up task event	Async event
AI Quality Monitor	Quality finding remediation task event	Async event
AI Concierge	Unresolved call follow-up task event; recovery follow-up task event	Async event
Lab Manager	SLA breach escalation event with task payload	Async event
Knowledge, Training & Learning	Pathway assignment event	Async event
HR & People Manager	Policy workflow step event; Critical Risk escalation event	Async event
Inventory & Compliance Manager	Compliance cycle task event with lifecycle state	Async event
Referral Manager	SLA breach escalation event; referral state follow-up event	Async event
Recall & Reconnect	Personal follow-up work item event	Async event
Financial Insights	Anomaly / risk follow-up task event	Async event
Product Shop	In-clinic fulfilment / exception-handling task event	Async event
Digital Forms	Form review / re-capture task event	Async event
Admin Control Plane	Tenant health event with task payload	Async event
Aiden	<code>AIAction</code> / <code>AIIntent</code> task suggestion ( <code>ActionType: Suggest</code> only; requires human confirmation)	Sync confirmation surface
Access Manager	Access revocation event	Async event

## 6.2 Outbound (this module emits to)

To module	What	Contract
Communication Hub	Task state-change notification triggers	Async event
Smart Dashboards	Open task queue state, overdue task counts, SLA state signals (near real-time)	Async event
Rota Manager	Read-only consumption of Work Package overlay; no write-back	—
Access Manager	RBAC enforcement queries	Sync
Audit & Compliance	Immutable audit event stream	Async event

### 6.3 PMS Boundary

Task Manager does not directly integrate with the Practice Management System (PMS). PMS-originated events that require operational follow-up reach Task Manager via intermediate modules (e.g. Appointment Manager, Communication Hub) rather than directly.

### 6.4 Rota Manager — Work Package Overlay Contract

Rota Manager defines `ShiftType: WorkPackage` as a first-class shift type. When a shift of this type is active, Rota Manager signals Task Manager via a Work Package overlay event. Task Manager MUST interpret this overlay as an instruction to surface and activate the work package(s) associated with that shift slot for the relevant role and location.

Specifically:

- Task Manager consumes the Work Package overlay event from Rota Manager, which carries at minimum the shift reference, role, location, and linked work package template identifier.
- Task Manager generates (or activates) the corresponding work package instance, consistent with its standard template-driven generation rules (idempotent, fully auditable, fresh instance per recurrence).
- Staff availability data that informs which roles are on shift remains the exclusive responsibility of Rota Manager; Task Manager does not replicate or override availability logic.
- The integration is read-only from Task Manager's perspective with respect to shift data: Task Manager receives the overlay event and acts on it; it does not write back to Rota Manager.

## 7. AI Boundaries (Non-Negotiable)

Module uses Aiden (Primoro AI Assistant) as an AI surface.

AI MAY:

- Suggest tasks or highlight risk to authorised staff for review (using `ActionType: Suggest` only when targeting Task Manager)
- Display `ConfidenceScore` from `AIIntent` to aid human judgement
- Summarise task activity for human review

## AI MAY NOT:

- Create, assign, or complete tasks autonomously
- Emit `ActionType` values of `Create`, `Assign`, or `Complete` when `TargetModule` is `task-manager`
- Bypass the `RequiresConfirmation: true` flag on any `AIAction` payload
- Override governance, audit, or access checks
- Make commitments on behalf of the practice
- Replace required clinical or managerial judgement

Task Manager MUST reject any `AIAction` payload where `RequiresConfirmation` is absent or `false`, or where `ActionType` is anything other than `Suggest` for Aiden-originated actions targeting this module. Confirmed AI-suggested tasks carry `Origin: AI-Suggested` and retain a `LinkedEntities` back-link to the originating `AIIntent`.

## 8. Audit & Compliance

---

The system MUST log all of the following events, immutably and with actor attribution and timestamp:

- Task and work package creation, including creation source (`Manual`, `Callback`, `Workflow`, `Template-Generated`, `AI-Suggested`), template linkage, and recurrence linkage
- Ownership assignment and reassignment (including system-initiated reversion to role queue on access revocation)
- All lifecycle state transitions, including Kanban board moves
- SLA state changes (`On-track` → `At-risk` → `Breached`)
- Checklist creation, checklist-item completion, and evidence capture (notes, comments, photos, file attachments)
- Task acknowledgement actions (distinct from claiming or completing)
- Completion and cancellation (with cancellation reason)
- Archival and retention policy application
- Secure-deletion events
- All AI suggestions received (`AIIntent` / `AIAction`), including which were accepted and which were rejected by authorised users, and including any payloads rejected for carrying disallowed `ActionType` values or absent `RequiresConfirmation` flags
- Access revocation events and resulting ownership reversion, recording the revoked user ID, timestamp, and new owner state

Audit logs MUST be immutable, attribution-complete, and exportable for inspection. Completed tasks retain their completion attribution to the named user even after that user's access is revoked; the audit log is never modified retrospectively.

## 9. Access Control

---

Access is enforced via Access Manager using role-based permissions.

The following permission boundaries apply:

Action	Who may perform it
--------	--------------------

Create task (manual)	Authorised staff per role permissions
Create task (event-driven / template)	System (on behalf of originating module)
Read task	Owner, assigned role, supervisors, administrators (confidential tasks: restricted)
Claim role-queue task	Eligible role members
Acknowledge task	Any user with read access to the task
Update task (progress, evidence)	Owner or claiming user
Reassign / change ownership	Supervisor or administrator role
Cancel task	Authorised staff per role permissions; requires reason
Archive task	System (retention policy); administrator
Administer templates and recurrence	Administrator role via Web Portal
Inspect audit log	Administrator / compliance officer (read-only)

Additional requirements:

- Device-level authentication is enforced on shared surgery and decontamination tablets via Access Manager.
- Access revocation propagates immediately across all devices, including forced logout of active sessions. Task Manager MUST revert ownership of in-flight tasks (Created, Active, In Progress, Blocked) to the associated role queue on receipt of a revocation event. Where no role queue can be determined, the task MUST be flagged for manual reassignment by an administrator and surfaced in the Web Portal exception view.
- Confidential tasks MUST restrict visibility to explicitly permitted roles and users.
- MFA requirements for sensitive operations (e.g. cancellation of compliance-linked tasks, bulk reassignment) are governed by Access Manager configuration and are not hardcoded within Task Manager.

## 10. Integration Summary

- **Rota Manager** — inbound Work Package overlay event (async); inbound AbsenceChangeEvent for absence-triggered tasks (async); read-only from Task Manager's perspective
- **Appointment Manager** — inbound AppointmentTaskTriggerEvent for lifecycle transition tasks (async); inbound Reschedule Job follow-up event (async)
- **Communication Hub** — inbound conversation-triggered task event (async); inbound message-to-work conversion task event (async); outbound task state-change notification triggers (async)
- **Aftercare Manager** — inbound aftercare trigger event (async)
- **Aiden** — inbound `AIAction` / `AIIntent` task suggestion surface (`ActionType: Suggest only`; sync confirmation)

- **AI Meeting Notes** — inbound confirmed action and retention review task events (async)
- **AI Guardian** — inbound finding follow-up task events (async)
- **AI Quality Monitor** — inbound quality finding remediation task events (async)
- **AI Concierge** — inbound unresolved call and recovery follow-up task events (async)
- **Lab Manager** — inbound SLA breach escalation event (async)
- **Knowledge, Training & Learning** — inbound pathway assignment event (async)
- **HR & People Manager** — inbound policy workflow step event (async); inbound Critical Risk escalation event (async)
- **Inventory & Compliance Manager** — inbound compliance cycle task event with lifecycle state mapping (async)
- **Referral Manager** — inbound SLA breach escalation event (async); inbound referral state follow-up event (async)
- **Recall & Reconnect** — inbound personal follow-up work item event (async)
- **Financial Insights** — inbound anomaly / risk follow-up task event (async)
- **Product Shop** — inbound in-clinic fulfilment / exception-handling task event (async)
- **Digital Forms** — inbound form review / re-capture task event (async)
- **Admin Control Plane** — inbound tenant health event (async)
- **Access Manager** — RBAC enforcement (sync); access revocation event (async)
- **Audit & Compliance** — outbound immutable audit event stream (async)
- **Smart Dashboards** — outbound open task queue state, overdue task counts, and SLA state signals (async, near real-time)

## 11. Explicit Non-Goals

---

- Not a replacement for full enterprise project-management tools; work-package depth is operational, not programme-management-grade.
- Does not store conversations or messages — owned by Communication Hub.
- Does not determine staff availability — owned by Rota Manager.
- Does not automate work without explicit human confirmation at every step.
- Not a clinical record system; clinical notes remain outside Task Manager's scope.
- Does not own Reschedule Job tracking for appointment rescheduling — owned by Appointment Manager.

## 12. Versioning & Governance

---

This specification is owned by: the Task Manager module owner.

Changes to this spec require:

- Review by the MVP module owner
- Impact analysis across all declared related modules (see /propose)
- Version bump (patch / minor / major depending on scope of change)

All future changes must preserve the following invariants:

- Calendar-backed work
- Explicit ownership at all times
- Fully auditable state transitions
- Role-first execution
- Separation of concerns between Task Manager and originating modules

## 13. Build Contract (Engineering & QA)

### 13.1 Canonical Data Model

#### Task

```

Task (
  TaskId          UUID PRIMARY KEY,          -- immutable
  Title           TEXT NOT NULL,
  Description     TEXT,
  Category       TEXT,
  DueDateTime    TIMESTAMPTZ,             -- mutually exclusive with TimeWindow fields
  TimeWindowStart TIMESTAMPTZ,
  TimeWindowEnd  TIMESTAMPTZ,
  Origin         ENUM('Manual','Callback','Workflow','Template-Generated','AI-Suggested') NOT NULL
  OwnerType     ENUM('User','Role') NOT NULL,
  OwnerId       UUID NOT NULL,
  Status        ENUM('Created','Active','In Progress','Blocked','Completed','Cancelled','Archived')
  SLAState      ENUM('On-track','At-risk','Breached'),
  Priority       TEXT,
  Confidential  BOOLEAN NOT NULL DEFAULT false,
  LinkedEntities JSONB,                   -- back-links to originating records
  Overdue       BOOLEAN NOT NULL DEFAULT false, -- computed and owned by Task Manager
  CompletedTimestamp TIMESTAMPTZ,
  CompletedByUserId UUID,
  CancellationReason TEXT,
  ArchivedAt    TIMESTAMPTZ,
  RetentionPolicyId UUID,
  AuditLog      JSONB NOT NULL           -- immutable append-only
)

```

#### Checklist

```

Checklist (
  ChecklistId      UUID PRIMARY KEY,          -- immutable
  ParentTaskId    UUID,                     -- FK to Task (nullable if parented to WorkPackage)
  ParentWorkPackageId UUID,                 -- FK to WorkPackage (nullable if parented to Task)
  AuditLog        JSONB NOT NULL
)

ChecklistItem (
  ChecklistItemId  UUID PRIMARY KEY,          -- immutable
  ChecklistId     UUID NOT NULL,             -- FK to Checklist
  Label          TEXT NOT NULL,
  CompletionState BOOLEAN NOT NULL DEFAULT false,
  Evidence       JSONB,                     -- notes, comments, photo/file references
  CompletedByUserId UUID,

```

```

CompletedTimestamp    TIMESTAMPTZ,
AuditLog              JSONB NOT NULL
)

```

## Work Package

```

WorkPackage (
  WorkPackageId      UUID PRIMARY KEY,          -- immutable
  Title              TEXT NOT NULL,
  Description         TEXT,
  TimeWindowStart    TIMESTAMPTZ NOT NULL,
  TimeWindowEnd      TIMESTAMPTZ NOT NULL,
  OwnerType          ENUM('User','Role') NOT NULL,
  OwnerId            UUID NOT NULL,
  Status             ENUM('Created','Active','In Progress','Blocked','Completed','Cancelled','Archived'),
  LinkedEntities     JSONB,
  AuditLog           JSONB NOT NULL
)

WorkPackageTask (
  WorkPackageId      UUID NOT NULL,
  TaskId             UUID NOT NULL,
  Sequence           INTEGER,
  PRIMARY KEY (WorkPackageId, TaskId)
)

```

## Templates & Recurrence

```

TaskTemplate (
  TaskTemplateId     UUID PRIMARY KEY,
  Title              TEXT NOT NULL,
  Description         TEXT,
  Category           TEXT,
  DefaultOwnerType   ENUM('User','Role'),
  DefaultOwnerId     UUID,
  DueWindowRules     JSONB,
  ChecklistStructure JSONB,
  EvidenceRequirements JSONB,
  AuditLog           JSONB NOT NULL
)

WorkPackageTemplate (
  WorkPackageTemplateId UUID PRIMARY KEY,
  Title                 TEXT NOT NULL,
  TaskTemplates         JSONB,
  ChecklistStructures   JSONB,
  AssignmentRules       JSONB,
  TimingRules           JSONB,
  AuditLog              JSONB NOT NULL
)

RecurrenceRule (
  RecurrenceRuleId   UUID PRIMARY KEY,
  TemplateId          UUID NOT NULL,
  TemplateType        ENUM('Task','WorkPackage') NOT NULL,
  Schedule             JSONB NOT NULL,          -- daily/weekly/monthly/custom definition
)

```

```
GeneratedInstances    JSONB,                -- links to generated instance IDs
AuditLog              JSONB NOT NULL
)
```

## 13.2 Core Behaviour Rules

1. Every task and work package MUST be calendar-backed: either `DueDateTime` or both `TimeWindowStart` and `TimeWindowEnd` MUST be set at creation.
2. Ownership MUST be explicit at all times; `OwnerId` and `OwnerType` MUST be non-null on every task.
3. All lifecycle state transitions — including Kanban board drag-and-drop moves — MUST be recorded in the immutable `AuditLog` with actor identity and timestamp.
4. Template recurrence MUST always generate a new instance per schedule tick; instances are never reused or silently duplicated (idempotent generation).
5. All evidence captured on checklist items (notes, comments, photos, files) MUST be attributable to a named user and timestamp.
6. Task Manager MUST reject any `AIAction` payload where `RequiresConfirmation` is absent or `false`, or where `ActionType` is anything other than `Suggest` for Aiden-originated actions targeting this module; AI is never authoritative.
7. On receipt of an access revocation event from Access Manager, Task Manager MUST revert ownership of all in-flight tasks (`Created`, `Active`, `In Progress`, `Blocked`) owned by the revoked user to the associated role queue. Where no role queue can be determined, the task MUST be flagged for manual reassignment.
8. The `Overdue` boolean flag MUST be computed and kept current by Task Manager; consuming surfaces MUST NOT compute it independently.
9. Acknowledgement MUST be recorded as a distinct, time-stamped, attributed audit event; it MUST NOT alter task lifecycle state or ownership.
10. All event-driven tasks from Admin Control Plane, AI Meeting Notes, AI Guardian, AI Quality Monitor, AI Concierge, Lab Manager, Knowledge Training & Learning, HR & People Manager, Inventory & Compliance Manager, Referral Manager, Recall & Reconnect, Financial Insights, Product Shop, Digital Forms, and Rota Manager (absence-triggered) MUST carry a `LinkedEntities` back-link to the originating record.
11. Cancellation MUST require and record a reason; the `CancellationReason` field MUST be non-null on any task in `Cancelled` state.
12. `Completed` and `Cancelled` tasks MUST NOT transition back to any operational state; they may only proceed to `Archived`.
13. Event-driven task creation MUST be idempotent; re-delivery of the same source event MUST NOT produce duplicate task instances.

## 13.3 Configuration Surfaces

- **Practice-level settings** (Admin Control Plane): default SLA windows, retention policy assignments, role-queue definitions, evidence requirements per category, confidential task visibility rules.
- **Per-user preferences** (Access Manager / Staff App Mode): personal queue display preferences, notification preferences.
- **Per-template overrides** (Task Manager Web Portal, administrator role): recurrence schedule, default ownership, due-window rules, checklist structure.
- **Per-work-package overrides** (Task Manager Web Portal, administrator role): claiming configuration, role assignment rules.

## 13.4 Filtering & Views

The UI MUST support the following standard filters across list, queue, and board views:

- Status (one or more lifecycle states)
- SLA state (On-track, At-risk, Breached)
- Owner / role
- Origin (Manual, Callback, Workflow, Template-Generated, AI-Suggested)
- Category
- Location
- Priority
- Confidential flag (where user has permission)
- Overdue flag
- Date range (due date / time window)

Standard views:

- **List views:** personal (user-owned), role/team, work-package with task drill-down
- **Queue views:** role-based, with claim and acknowledge actions
- **Board (Kanban) view:** tasks grouped by lifecycle state; personal, role-scoped, and location-scoped boards; all moves are audited state transitions; no new lifecycle states introduced by the board view

## 13.5 Module Extension Map

Future extensions MUST preserve the following invariants to avoid breaking this contract:

- The `Origin` enum may be extended with new values; existing values are frozen.
- The `Status` state machine may add intermediate states only between `Active` and `Completed`; terminal states (`Completed`, `Cancelled`, `Archived`) are frozen.
- `LinkedEntities` is a JSONB field; new back-link types may be added without schema migration.
- New event-driven sources integrate by emitting a payload conforming to the existing event-driven task contract (§4.3); no core schema changes are required.
- Template structures use JSONB for extensibility; new template fields may be added non-breaking.

## 13.6 Acceptance Criteria

The build of Task Manager is complete when:

- [ ] All canonical objects (`Task`, `Checklist`, `ChecklistItem`, `WorkPackage`, `TaskTemplate`, `WorkPackageTemplate`, `RecurrenceRule`) can be created, read, and updated through the API
- [ ] State machine transitions enforce all rules in §3.2 and §13.2, including terminal-state freezing and cancellation-reason requirement
- [ ] Kanban board drag-and-drop is treated as an audited state transition identical to programmatic transitions
- [ ] All integrations in §6 are wired and event payloads validated
- [ ] All boundaries in §7 are enforced; negative tests confirm `AIAction` payloads without `RequiresConfirmation: true` are rejected, and payloads with `ActionType` of `Create`, `Assign`, or

Complete from Aiden are rejected

- [ ] Audit log captures every event class in §8 with actor and timestamp
- [ ] Access control is enforced per §9; confidential task visibility is restricted correctly
- [ ] Access revocation from Access Manager triggers role-queue ownership reversion on all in-flight tasks (§8, §9)
- [ ] Acknowledgement action is available as a distinct control in Staff App Mode and recorded as a distinct audit event
- [ ] Overdue flag is computed by Task Manager and surfaced correctly in the tablet data contract fields (DueTime, Status, CompletedTimestamp, CompletedByUserId, Overdue)
- [ ] AI Meeting Notes task creation works for both confirmed action tasks and retention review tasks, with correct LinkedEntities back-links and idempotent behaviour on event re-delivery
- [ ] AI Guardian finding follow-up task creation contract is satisfied, including LinkedEntities back-link
- [ ] AI Quality Monitor quality finding remediation task creation contract is satisfied, including LinkedEntities back-link
- [ ] AI Concierge unresolved call and recovery follow-up task creation contract is satisfied
- [ ] Lab Manager escalation task creation contract is satisfied, including LinkedEntities back-link and default ownership
- [ ] Knowledge, Training & Learning pathway assignment task creation is functional with lifecycle tracking
- [ ] HR & People Manager workflow step and Critical Risk escalation task creation carry correct Origin: Workflow and LinkedEntities
- [ ] Inventory & Compliance Manager lifecycle state mapping is implemented per the table in §4.3
- [ ] Referral Manager SLA breach and referral state follow-up task creation contracts are satisfied
- [ ] Recall & Reconnect personal follow-up work item creation is functional with correct LinkedEntities back-link
- [ ] Financial Insights anomaly / risk follow-up task creation is functional
- [ ] Product Shop in-clinic fulfilment / exception-handling task creation is functional
- [ ] Digital Forms form review / re-capture task creation is functional
- [ ] Rota Manager AbsenceChangeEvent consumption triggers absence-driven tasks correctly where configured
- [ ] Appointment Manager AppointmentTaskTriggerEvent consumption creates lifecycle transition tasks correctly
- [ ] Admin Control Plane health event task creation is functional and tenant-scoped
- [ ] Rota Manager Work Package overlay event is consumed and activates the correct work package instance idempotently
- [ ] Template recurrence generates new instances idempotently per schedule, with LinkedEntities back-link to template
- [ ] Smart Dashboards outbound signal (open task queue, overdue counts, SLA state) is emitted correctly on state change
- [ ] Staff App Mode task surface applies correct role-scoped and location-scoped filtering rules
- [ ] All non-functional requirements in §14 are met

## 14. Non-Functional Requirements

---

- **Performance:** Board and list views MUST load within acceptable latency under expected concurrent user load. `Overdue` flag MUST be recomputed without perceptible lag at the point of due-time expiry.
- **Reliability:** No silent loss of task state or audit events. Idempotent recurrence generation MUST be safe to retry without creating duplicate instances. Graceful degradation is required: if an upstream event source is unavailable, Task Manager MUST queue incoming events and process them on recovery rather than silently discarding them.
- **Scalability:** The module MUST support multi-site, multi-tenant deployments. Role queues, board views, and tablet surfaces MUST be correctly scoped by location and tenant without cross-tenant data leakage.
- **Security:** All task data MUST be encrypted at rest and in transit. Secrets and key management follow platform conventions. Confidential task payloads MUST not be exposed to unauthorised roles at any layer, including event payloads emitted to other modules.
- **Privacy:** Task data linked to patients (e.g. callback tasks) is subject to GDPR data-subject rights. Retention policies and secure-deletion events MUST be audit-logged. Data retention periods are configurable per practice via Admin Control Plane; Task Manager enforces the configured policy but does not set default retention periods unilaterally.
- **Observability:** Task Manager MUST export metrics covering task creation rate, SLA breach rate, overdue task count by role and location, and event-processing latency. Structured logs MUST be emitted for all state transitions. Distributed traces MUST be propagated across cross-module event boundaries to support end-to-end latency diagnosis.
- **Accessibility:** All staff-facing UI surfaces (Web Portal, Staff App Mode, tablet views) MUST meet WCAG 2.1 AA accessibility standards, consistent with platform-wide requirements.

## 15. Open Questions

---

Outstanding decisions before this spec can be promoted from `draft` to `published`.

1. **Confidential task event payloads** — When Task Manager emits a state-change event to Communication Hub for a confidential task, what data should the payload contain? The original is silent on whether confidential task titles/descriptions are redacted in outbound events.
2. **Compliance lifecycle state mapping agreement** — The spec states that Inventory & Compliance Manager and Task Manager MUST agree the compliance lifecycle state mapping during integration design. This agreement has not yet been completed; the mapping table in §4.3 is a draft proposal pending sign-off from both module owners.
3. **Retention periods** — The spec references configurable retention policies and secure deletion but does not define default retention periods for any task category. These defaults need to be defined before the Admin Control Plane configuration surface can be built.
4. **Overdue flag recomputation mechanism** — The spec requires the `Overdue` flag to be computed and kept current by Task Manager, but does not specify whether this is event-driven (triggered at `DueTime`), polling-based, or a read-time computation. The implementation approach needs to be agreed before the tablet data contract acceptance criterion can be validated.
5. **Board view scope for work packages** — The spec states that personal, role-scoped, and location-scoped boards are supported. It is silent on whether a board view scoped to a work package (showing constituent tasks) is also required. This needs clarification before UI build.

**6. Absence-driven task creation configuration** — The spec notes that Rota Manager `AbsenceChangeEvent` MAY trigger task creation, depending on integration configuration. The specific absence state transitions that should trigger tasks (and their default payloads) have not yet been agreed between the Rota Manager and Task Manager module owners.

**7. Smart Dashboards signal contract** — The outbound signal contract for Smart Dashboards (open task queue state, overdue task counts, SLA state) is described at a high level. The precise payload schema, polling interval or push mechanism, and per-role filtering semantics need to be defined before the Smart Dashboards integration acceptance criterion can be validated.