

Appointment Manager

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

Appointment Manager – Technical Specification

1. Module Purpose & Scope (Authoritative)

The Appointment Manager is the governed scheduling engine and single system of record for all appointment booking and diary availability within Primoro. Its purpose is to ensure appointments are booked correctly the first time, constrained by explicit clinical, operational, accessibility, rota, zone, and policy rules, and are consistent across staff, patient, and AI-assisted journeys. All appointments are calendar-backed events within Primoro CORE and must never be free-form diary entries.

It governs:

- End-to-end booking, rescheduling, and cancellation of appointments (staff- and patient-initiated), including constraint enforcement across booking windows, rota availability, practitioner eligibility, surgery eligibility, zone and zone-group rules, and accessibility and additional needs
- Availability publication for diary views, Find Next, and patient self-booking — sourced exclusively from Rota Manager-generated Rota Entries
- Digital Waitlist operation for cancellation backfill, including offer lifecycle management and staff-facing automation visibility
- Care plan entitlement validation at booking time, deposit policy rule ownership, and emission of deposit collection events to Integrated Payments
- The appointment lifecycle state machine from creation through confirmation, arrival, completion, no-show, and cancellation, with a complete immutable audit trail
- Governed integration interfaces for Lab Manager (appointment-linkage), Surgery & Decon Day View (today-only operational feed), Task Manager (task-trigger events), AI Concierge (short-notice cancellation events), and Smart Dashboards (read-only data exposure)
- Emission of appointment lifecycle signals to downstream consumers including Recall & Reconnect, Rewards Manager, Aftercare Manager, Performance Dashboards, AI Guardian, and Financial Insights — in each case Appointment Manager is the authoritative source of lifecycle state; downstream modules are responsible for their own processing logic

Appointment Manager also serves as the governed availability surface for AI-assisted booking journeys. AI Assistant (Aiden) and AI Concierge MAY guide patients towards available slots, but Appointment Manager remains the sole arbiter of availability and booking validity. These surfaces are delivery channels for Appointment Manager's availability, not decision-makers in their own right.

Booking creation is subject to a pre-condition gate: Appointment Manager MUST validate that Treatment Pipeline Manager has set `BookingEligibility = true` for the patient before permitting any new appointment to be created. This prevents leakage, duplication, and unsafe automation across all intake channels.

It explicitly does not:

- Own message delivery — that is the responsibility of Communication Hub

- Own schedule pattern creation or management — that is the responsibility of Rota Manager
- Own task creation, assignment, or execution — that is the responsibility of Task Manager
- Own clinical notes or diagnosis — that is the responsibility of the PMS
- Own payment settlement or ledger logic — that is the responsibility of Integrated Payments

2. Ownership & Responsibilities

2.1 Appointment Manager IS Responsible For

- Booking, rescheduling, and cancelling appointments (staff- and patient-initiated)
- Enforcing all booking constraints: booking windows, rota availability derived exclusively from Rota Manager-generated Rota Entries, practitioner eligibility, surgery eligibility, zone and zone-group constraints, and accessibility and additional needs
- Validating `BookingEligibility = true` as set by Treatment Pipeline Manager before permitting appointment creation; booking MUST be rejected if this field is absent or false
- Validating care plan entitlement at booking time — confirming active plan membership, enforcing waiting-period and minimum-contribution thresholds, confirming sufficient remaining entitlement, and marking covered services as £0 on the invoice record passed downstream at confirmation; this includes consuming and displaying the `entitlement_status` and `unlock_date` fields exposed by Care Plan Subscriptions to provide contextual eligibility clarity during the booking journey
- Enforcing cadence-locking signals from Care Plan Subscriptions: where Care Plan Subscriptions emits a `cadence_locked` signal for a plan member, Appointment Manager MUST validate against this signal during appointment creation and rescheduling to prevent booking within locked recall intervals
- Owning deposit policy rules and emitting `DepositCollectionEvent` to Integrated Payments
- Exposing a queryable appointment-linkage interface for Lab Manager
- Publishing a today-only, rota-context-filtered appointment feed for operational downstream surfaces including Surgery & Decon Day View
- Exposing a governed read-only data interface for external consumers including Smart Dashboards
- Emitting `AppointmentTaskTriggerEvent` to Task Manager when a lifecycle transition creates a downstream task obligation
- Emitting `RotaCancellationEvent` to AI Concierge on short-notice cancellations
- Operating the Digital Waitlist for cancellation backfill and exposing staff-facing automation state
- Managing appointment types as the primary scheduling control surface
- Producing a complete immutable audit trail for all lifecycle events, automation actions, and integration emissions
- Emitting appointment lifecycle signals (booking confirmed, completed, cancelled, no-show) to Recall & Reconnect, Rewards Manager, Aftercare Manager, Performance Dashboards, AI Guardian, and Financial Insights via governed outbound events

2.2 Appointment Manager IS NOT Responsible For

- Message delivery — owned by Communication Hub
- Schedule pattern management — owned by Rota Manager
- Task creation, assignment, and execution — owned by Task Manager

- Clinical notes or diagnosis — owned by the PMS
- Payment settlement, payment retries, or ledger logic — owned by Integrated Payments
- Autonomous decision-making by AI
- Deposit policy rule ownership — this is explicitly owned by Appointment Manager (not Integrated Payments); Integrated Payments handles only the downstream settlement and ledger operations triggered by `DepositCollectionEvent`
- Recall interval calculation or next-due date derivation — owned by Recall & Reconnect (which consumes last-completed-appointment signals from Appointment Manager)
- Referral workflow management — owned by Referral Manager (which links to appointments via the appointment-linkage interface)
- Leave approval workflows — owned by HR & People Manager (Appointment Manager consumes only confirmed approved-leave state transitions, not draft or submitted leave states)

3. Core Objects (Normative)

3.1 Appointment (Canonical Artefact)

An Appointment is a governed digital artefact representing a confirmed or in-progress calendar-backed scheduling commitment between a patient and a practitioner within a defined surgery context.

Minimum required fields:

- `AppointmentID` — unique identifier
- `PatientID` — foreign key to patient record
- `PractitionerID` — assigned practitioner
- `SurgeryID` — assigned surgery room
- `AppointmentTypeID` — driving scheduling constraints
- `RotaEntryID` — foreign key to the authoritative Rota Entry that generated the slot
- `LifecycleState` — current state (see §3.2)
- `BookingSource` — `staff` | `patient` | `system`
- `CreatedBy` — user or role identity
- `CreatedAt` — UTC timestamp
- `AuditTrail` — immutable append-only log of all state transitions and events
- `ToothNumbers` — array of tooth identifiers relevant to the appointment (nullable; populated where clinically applicable; exposed to Surgery & Decon Day View via the operational feed)
- `LabRequired` — boolean flag indicating whether a lab case is associated with this appointment (nullable; sourced from or synchronised with Lab Manager)
- `LabCaseReference` — reference identifier linking to the associated Lab Manager lab case (nullable; read-only from Appointment Manager's perspective once set by Lab Manager)

`LabRequired` and `LabCaseReference` are owned by Lab Manager. Appointment Manager stores them as linked read-only properties and MUST NOT modify them directly. They are surfaced in the operational feed (§14) to support Surgery & Decon Day View rendering requirements. `ToothNumbers` is authored within Appointment Manager or the PMS and passed through to the operational feed.

3.2 Appointment State Machine (Authoritative)

States:

- `created` — appointment record exists; not yet confirmed
- `confirmed` — booking confirmed; deposit event emitted if applicable
- `arrived` — patient checked in
- `in_progress` — appointment session active (used as the basis for Aftercare Manager triggers and real-time current-time rendering in Day View)
- `completed` — appointment concluded
- `no-show` — patient did not attend
- `cancelled` — appointment cancelled by staff, patient, or system

Rules:

- All state transitions are auditable and time-stamped with actor identity
- An appointment in `completed`, `no-show`, or `cancelled` state MUST NOT return to any earlier state
- Cancellation from any pre-completed state is permitted, subject to access control
- Transition from `confirmed` to `cancelled` by the system (e.g. practitioner absence) MUST generate a Reschedule Job (§8)
- Transition to `cancelled` within the configured short-notice window MUST emit a `RotaCancellationEvent` to AI Concierge (§10.1)
- Transition to `completed`, `no-show`, or `cancelled` MUST emit an `AppointmentTaskTriggerEvent` to Task Manager (§17) where a task is configured on the Appointment Type
- Transition to `completed` MUST emit an `AppointmentCompletionEvent` to Aftercare Manager, Recall & Reconnect, Rewards Manager, and any other registered downstream consumer; this event constitutes the authoritative signal that an appointment session has concluded
- Transition to `completed` or `no-show` MUST emit the relevant lifecycle signal to Performance Dashboards for DNA and utilisation metric calculation (§11.2)
- **Digital Forms gate:** Transition from `in_progress` to `completed` via Surgery Tablet Finish & Send MUST NOT proceed if Digital Forms reports one or more mandatory unsigned forms remain for that appointment. Appointment Manager MUST query Digital Forms for form-completion status before permitting this transition. If the gate blocks completion, the attempt MUST be rejected with an explicit reason and logged in the audit trail.

3.3 Reschedule Job (Canonical Artefact)

A Reschedule Job is a governed automation artefact created by Appointment Manager when practitioner absence invalidates one or more confirmed appointments.

Minimum required fields:

- `RescheduleJobID` — unique identifier
- `TriggerAbsenceSource` — reference to the Rota Manager absence record that caused creation
- `AffectedAppointmentIDs` — list of appointment identifiers in scope
- `PerPatientStatus` — independent tracking record per affected patient
- `CreatedAt` — UTC timestamp

- `AuditTrail` — immutable log of job progress and manual interventions

Tasks generated by Appointment Manager's reschedule flow are emitted via `AppointmentTaskTriggerEvent` (§16) and owned by Task Manager upon receipt. If Access Manager revokes a user's access whilst reschedule-related tasks are in flight, Task Manager is responsible for reverting those task ownerships to appropriate role queues. Appointment Manager has no obligation to track or manage task ownership transitions following event emission.

3.4 Waitlist Offer (Canonical Artefact)

A Waitlist Offer is a governed automation artefact representing an opportunity to fill a cancelled slot, issued to one or more eligible patients via the Digital Waitlist.

Minimum required fields:

- `OfferID` — unique identifier
- `CancelledAppointmentID` — source slot
- `OfferState` — current state (see §7.2)
- `EligiblePatientIDs` — patients broadcast to
- `LockHeldByPatientID` — patient currently holding engagement lock (nullable)
- `LockExpiresAt` — UTC timestamp of lock expiry (nullable)
- `CreatedAt` — UTC timestamp
- `AuditTrail` — immutable log of offer events and outcomes

4. Availability & Scheduling Constraints

4.1 Availability Source of Truth (Authoritative)

Appointment Manager consumes availability only from Rota Entry records generated and supplied by Rota Manager. Rota Manager is the single system of record for staff availability and the authoritative availability input layer for Appointment Manager. Appointment Manager MUST NOT derive or infer availability from any other source, including HR systems, historical bookings, or treatment-type filters.

Rota Entry records consumed by Appointment Manager include generated entries from Schedule Patterns, manually authored entries, and absence/leave overrides, all as produced and exported by Rota Manager. Appointment Manager MUST consume rota entries without filtering by treatment type; availability is role-based, not treatment-based, consistent with Rota Manager's governing principle.

Appointment Manager MUST NOT:

- Read or interpret schedule patterns directly
- Infer availability from historical bookings
- Generate availability independently of Rota Manager
- Filter rota entries by treatment type when resolving availability

Availability is authoritative only when a valid Rota Entry exists and is not overridden by absence, leave, or break segments.

4.2 Break Segment Enforcement (Authoritative)

Rota Manager generates RotaEntry records with a `ShiftType` field. Where `ShiftType = Break`, the entry represents protected non-bookable time.

The module MUST:

- Identify all RotaEntry records carrying `ShiftType = Break`
- Treat the time ranges they cover as unavailable blocks
- Never surface break-typed RotaEntry ranges as bookable slots in any availability search, diary view, or self-booking journey

Break blocks MUST NOT be overridable by Appointment Type configuration, staff action, or AI suggestion. They are structurally equivalent to absence and leave overrides for the purposes of availability resolution.

4.3 Availability Search Model (Authoritative)

All availability searches MUST be bounded by the intersection of:

Booking Window \cap Rota Entry \cap Practitioner \cap Surgery \cap Zone \cap Appointment Type \cap Patient Needs

The entire diary MUST NOT be searched.

When no availability exists, the system MUST:

- Explicitly explain why no availability was found
- Offer governed alternatives where applicable (e.g. Digital Waitlist enrolment, Find Next results)

4.4 Clinician Substitution Rules

- Dentist appointments MUST default to the same clinician
- Hygiene appointments MAY allow clinician substitution where configured
- Emergency appointments MAY prioritise earliest availability

Substitution rules are derived from Appointment Type configuration.

5. Delivery Surfaces & Access (Authoritative)

5.1 Web Portal (Staff)

Appointment Manager renders through a single canonical calendar engine exposed on the staff web portal. Two lenses are available:

- Practitioner-first (default)
- Surgery-first (toggle)

Lens changes re-render the same underlying data model and MUST NOT trigger availability recomputation.

Day, Week, and Month views share date and filter state; switching views MUST NOT reset selection or context. Month view is non-interactive (decision support only).

5.2 Tablet App

Appointment Manager surfaces appointment lookup, lifecycle state updates (arrived, completed, no-show), and today's schedule on the in-practice tablet, subject to the same RBAC constraints as the web portal.

(Detailed tablet interaction patterns are governed by the Appointment Manager – UX Specification.)

5.3 Patient Mobile App

Appointment Manager exposes patient self-booking subject to governed availability (§4.3), care plan entitlement validation, and deposit policy rules. Patients may view upcoming appointments and initiate cancellation within configured policy windows.

5.4 Real-Time Current-Time Rendering (Day View)

The system MUST support a real-time current-time rendering layer in Day View.

Requirements:

- Render a single horizontal "now" line across the full diary grid
- Be time-zone aware and driven by the local practice clock
- Update continuously without full diary re-render or availability recomputation
- Remain correctly positioned during vertical scroll

When the current time intersects an appointment, the system MUST expose rendering metadata to allow the UI to identify and visually emphasise the active appointment. Elapsed vs upcoming time MAY be visually differentiated via presentation state only.

Rules:

- Presentational only
- No effect on booking, arrival, or audit logic
- MUST NOT be treated as a workflow trigger

5.5 Engagement Signals

Appointment Manager emits:

- Real-time offer state for the Offer Monitor (§7.4)
- Real-time reschedule job progress for the Reschedule Job Monitor (§8.3)
- Today's appointment feed for Surgery & Decon Day View (§15)
- Live appointment state for Smart Dashboards via the read-only data interface (§16)

6. Find Next Appointment Finder

Find Next is the primary governed availability discovery surface.

It MUST:

- Operate as a persistent modal overlay
- Preserve decision context across interaction steps
- Return only compliant availability bounded by the search model in §4.3
- Rank results without auto-selection

Booking from Find Next requires explicit confirmation. Find Next MUST NOT auto-book on behalf of a patient or staff member.

7. Digital Waitlist (Governed Automation)

7.1 Purpose

The Digital Waitlist converts cancellations into a governed opportunity to fill unused chair time while preserving fairness and constraint integrity. Staff observe automation state; the system issues offers.

7.2 Offer Lifecycle (Authoritative)

Each Waitlist Offer follows the lifecycle:

- `broadcast_offered` — slot offered to all eligible patients simultaneously
- `engaged_locked` — a patient engages; slot is temporarily locked for that patient
- `confirmed` — booking (and any required deposit) completed; offer is closed
- `expired` — engagement window elapsed without confirmation; slot returns to the available pool

Only one confirmed booking may result from a single offer.

7.3 Engagement Lock Window

- Engagement lock duration is finite and short; the duration is practice-configurable
- A slot **MUST** be locked only after patient engagement
- Failure to complete confirmation or deposit payment within the lock window results in automatic expiry and slot return to the pool
- **Delegated users (Family Profiles):** Where a patient is represented by a guardian or delegated user via Family Profiles, that delegated user **MUST** respond to waitlist offers within the same engagement lock window that applies to direct patients. No extended window is granted for delegated engagement. The engagement lock is bound to the offer and slot, not to the engagement channel or user type.

Indefinite reservation states are prohibited. Lock window configuration **MUST** have a defined minimum and maximum permissible range, set during implementation and documented in the deployment configuration guide.

7.4 Offer Monitor (Staff Visibility Contract)

Appointment Manager **MUST** expose a staff-facing Offer Monitor showing real-time automation state.

The monitor **MUST**:

- Show one row per cancelled slot
- Surface offer state, timing, and outcome
- Distinguish system actions from human intervention
- Prioritise active and stalled offers

The Offer Monitor is observational, not a control surface.

8. Practitioner Absence–Driven Rescheduling

8.1 Source of Authority

All practitioner absence or leave originates in Rota Manager. Appointment Manager consumes confirmed rota changes and executes downstream effects.

Specifically, the canonical trigger for absence-driven rescheduling is Rota Manager's `AbsenceChangeEvent`, emitted when absence is confirmed and availability is immediately removed. Appointment Manager **MUST NOT**

initiate rescheduling logic from HR feeds, leave-request drafts, or submitted-but-unapproved leave states. Only confirmed, approved absence or leave events — as propagated through Rota Manager — constitute a valid trigger.

Where leave is originated and approved via HR & People Manager's leave approval workflow, the approved state transition is first validated against rota availability by HR & People Manager before Rota Manager confirms the absence. Appointment Manager consumes only the resulting confirmed `AbsenceChangeEvent` from Rota Manager; it has no direct dependency on HR & People Manager's internal leave state machine.

8.2 Reschedule Job Model

When absence impacts booked appointments, Appointment Manager MUST create a Reschedule Job. Each affected patient is tracked independently within the job's per-patient status record.

8.3 Reschedule Job Monitor (Staff Visibility)

The system MUST provide a staff-visible Reschedule Job Monitor that:

- Shows job-level progress
- Shows per-patient status
- Allows proportionate manual takeover per patient

Invoking manual handling pauses automation for that patient only; it does not affect the job or other patients within the same job.

9. Access Control

Access to Appointment Manager capabilities is governed by Access Manager RBAC roles.

The following capability-level access rules apply:

Capability	Minimum Role Required
View diary and appointment detail	Reception / Clinical Staff
Create and confirm appointments (staff-initiated)	Reception
Reschedule or cancel appointments (staff-initiated)	Reception
Access Offer Monitor	Reception / Practice Manager
Access Reschedule Job Monitor and trigger manual takeover	Practice Manager
Read-only data interface (Smart Dashboards)	Authorised external consumer, context-scoped
Appointment Type configuration	Practice Manager / Admin
Deposit policy configuration	Practice Manager / Admin

The above role names are indicative. Authoritative role definitions live in Access Manager. This section must be reconciled with Access Manager's role registry before the build contract is promoted to published.

MFA requirements for sensitive operations (e.g. bulk cancellation, absence-driven reschedule job initiation) are to be confirmed with the Access Manager specification.

10. AI Boundaries (Non-Negotiable)

AI MAY:

- Explain availability, eligibility, or constraint reasons to patients or staff
- Summarise automation progress for staff review
- Present recovery options to patients following a short-notice cancellation event (via AI Concierge, per §10.1)

AI MAY NOT:

- Auto-book any appointment
- Auto-cancel any appointment
- Override booking constraints, deposit policy, or care plan entitlement rules
- Bypass governance, audit, or access checks
- Make commitments on behalf of the practice

All AI output MUST be advisory, transparent, and auditable. Any AI suggestion surface MUST log the suggestion and whether it was accepted or rejected by a human actor.

AI Assistant (Aiden) and AI Concierge are delivery surfaces for Appointment Manager's governed availability. They guide patients or staff towards available slots but MUST NOT own or influence the availability computation or booking validation logic. All booking requests originating from AI-assisted flows MUST pass through Appointment Manager's standard governed booking flow, subject to all applicable constraints.

Booking requests that originate from accepted Smart Treatment Proposals are coordinated by Communication Hub and surfaced as standard booking intents to Appointment Manager. Appointment Manager MUST NOT treat proposal acceptance state as a booking pre-condition or bypass; it processes the booking request on its own merits against all applicable constraints.

10.1 AI Concierge – Short-Notice Cancellation Event Contract

When an appointment is cancelled and the cancellation falls within a configurable short-notice window, Appointment Manager MUST emit a `RotaCancellationEvent` to AI Concierge.

Short-notice window: The threshold (expressed as minutes before appointment start time) is practice-configurable and stored as a module-level parameter. The default value MUST be defined during implementation and documented in the deployment configuration guide.

Event payload MUST include:

Field	Description
<code>appointmentId</code>	Unique identifier of the cancelled appointment
<code>patientId</code>	Identifier of the affected patient

<code>practitionerId</code>	Identifier of the originally assigned practitioner
<code>appointmentTypeId</code>	Identifier of the appointment type
<code>originalSlotStart</code>	UTC timestamp of the cancelled slot start
<code>originalSlotEnd</code>	UTC timestamp of the cancelled slot end
<code>cancellationTimestamp</code>	UTC timestamp at which cancellation was confirmed
<code>shortNoticeWindowMinutes</code>	The configured threshold that caused classification as short-notice
<code>cancellationSource</code>	<code>staff</code> <code>patient</code> <code>system</code>

AI Concierge handling rules:

- AI Concierge MAY present recovery options to the patient or escalate to staff; it MUST NOT auto-book
- If a patient accepts a recovery slot via AI Concierge, the booking handoff MUST be routed through Appointment Manager's standard governed booking flow, subject to all applicable constraints (booking window, practitioner eligibility, zone, accessibility, care plan entitlement, and deposit rules)
- The original `appointmentId` MUST be referenced in the replacement booking's audit record to preserve traceability

The `RotaCancellationEvent` is distinct from the standard reschedule job flow (§8) and applies specifically to short-notice cancellations where proactive patient engagement is appropriate.

11. Integration Contracts

11.1 Inbound (this module consumes from)

From Module	What	Contract
Rota Manager	RotaEntry records (including <code>ShiftType</code>), confirmed absence/leave changes, and <code>AbsenceChangeEvent</code> on absence confirmation	Sync / event-driven
Access Manager	RBAC role assertions for all capability access checks	Sync
Treatment Pipeline Manager	<code>BookingEligibility</code> field — MUST be <code>true</code> before any appointment creation is permitted	Sync (query at booking time)

Care Plan Subscriptions	<code>entitlement_status</code> , <code>unlock_date</code> , and <code>cadence_locked</code> signals for plan members — consumed at booking time for eligibility display and interval enforcement	Sync (query at booking time)
Digital Forms	Form-completion status for mandatory forms — queried before permitting <code>in_progress</code> → <code>completed</code> transition via Surgery Tablet Finish & Send	Sync (query at transition time)
Communication Hub	CallThread hand-off context — booking creation or scheduling flows may be initiated from a call context via Communication Hub; Appointment Manager receives the booking intent and processes it through standard governed flow	Event / async
HR & People Manager	Confirmed approved-leave state transitions — consumed indirectly via Rota Manager's <code>AbsenceChangeEvent</code> ; no direct HR feed dependency	(via Rota Manager)

11.2 Outbound (this module emits to)

To Module	What	Contract
Communication Hub	Notification triggers on offer state transitions and reschedule actions	Event (async)
Integrated Payments	<code>DepositCollectionEvent</code> on booking confirmation where deposit policy applies	Event (async)
Task Manager	<code>AppointmentTaskTriggerEvent</code> on qualifying lifecycle transitions	Event (async)
AI Concierge	<code>RotaCancellationEvent</code> on short-notice cancellations	Event (async)
Lab Manager	Appointment-linkage query responses (read-only, query-response)	Sync

Surgery & Decon Day View	Today-only, rota-context-filtered appointment feed	Real-time read feed
Smart Dashboards	Live appointment state via governed read-only data interface	Governed read (sync/near-real-time)
Audit & Compliance	Immutable event log entries for all auditable actions	Event (async)
Recall & Reconnect	<code>AppointmentCompletionEvent</code> on transition to <code>completed</code> ; last-completed-appointment date for recall interval calculation	Event (async)
Rewards Manager	<code>AppointmentAttendanceEvent</code> on transition to <code>completed</code> — consumed by Rewards Manager for attendance-based point accrual and referral reward issuance	Event (async)
Aftercare Manager	<code>AppointmentCompletionEvent</code> on transition to <code>completed</code> — consumed by Aftercare Manager as the authoritative trigger for aftercare workflow initiation	Event (async)
Performance Dashboards	Lifecycle events (booking confirmed, cancellation, <code>completed</code> , <code>no-show</code>) including slot utilisation context — consumed for real-time metric calculation (see §11.3)	Event (async)
AI Guardian	Booking, completion, and cancellation signals — consumed by AI Guardian as upstream appointment signal source	Event (async)
Financial Insights	Booking creation, modification, and cancellation signals with practitioner and appointment-type context — consumed for production and revenue attribution	Event (async)

Referral Manager	Appointment-linkage query responses for referral-linked appointments (read-only, query-response) — Referral Manager may query appointment state to determine whether a referral-linked appointment exists and is valid	Sync
------------------	--	------

11.3 Performance Dashboards – Lifecycle Event Contract

Performance Dashboards consumes real-time lifecycle events from Appointment Manager for booking, cancellation, DNA (did-not-attend / no-show), and utilisation metric calculation. Appointment Manager MUST emit these events with sufficient context for metric attribution.

Minimum event payload fields for Performance Dashboards lifecycle events:

Field	Description
appointmentId	Unique identifier of the appointment
patientId	Patient identifier
practitionerId	Assigned practitioner at time of event
surgeryId	Surgery room
appointmentTypeId	Appointment type
lifecycleTransition	The transition that occurred (confirmed, completed, no-show, cancelled)
transitionTimestamp	UTC timestamp of the transition
slotStart	UTC timestamp of the slot start
slotEnd	UTC timestamp of the slot end
cancellationSource	staff patient system (present on cancellation events only)
bookingSource	staff patient system (present on confirmation events only)

Performance Dashboards is responsible for all metric aggregation and attribution logic. Appointment Manager's responsibility ends at event emission.

11.4 PMS Boundary

The PMS owns clinical notes, diagnosis, and treatment records. Appointment Manager owns the scheduling event and passes invoice-relevant entitlement data (covered service marked £0) downstream at confirmation. The PMS MUST NOT generate or modify appointment lifecycle state directly; Appointment Manager is the system of record for all scheduling state.

12. Deposit Collection Event Contract

Appointment Manager owns deposit policy rules. When a booking confirmation requires a deposit, Appointment Manager MUST emit a `DepositCollectionEvent` to Integrated Payments.

Trigger conditions:

- A new appointment is confirmed and the applicable Appointment Type carries a deposit policy
- A rescheduled appointment resets deposit eligibility per policy rules

Event payload MUST include:

Field	Description
<code>appointmentId</code>	Unique identifier of the booked appointment
<code>patientId</code>	Patient against whom the deposit is to be collected
<code>depositAmountPence</code>	Deposit amount in pence as derived from the policy
<code>appointmentTypeId</code>	Appointment type driving the deposit rule
<code>bookingTimestamp</code>	UTC timestamp of the confirmed booking
<code>policyRuleId</code>	Identifier of the deposit policy rule applied
<code>carePlanCovered</code>	Boolean — <code>true</code> if care plan entitlement has reduced or eliminated the deposit obligation

Integrated Payments is responsible for all subsequent payment processing and ledger logic. Appointment Manager MUST NOT perform or retry payment operations. Appointment Manager's responsibility ends at event emission.

13. Lab Manager – Appointment-Linkage Interface

Lab Manager requires the ability to verify whether a specific follow-up appointment exists and is in a valid state before a Lab Case may advance to Received status. Referral Manager similarly requires the ability to verify whether a referral-linked appointment is valid.

Appointment Manager MUST expose a queryable appointment-linkage interface that:

- Accepts a `labCaseId`, `appointmentId`, or `referralId` reference
- Returns the current lifecycle state of the linked appointment (`created`, `confirmed`, `arrived`, `completed`, `cancelled`, `no-show`)

- Indicates whether the appointment qualifies as a valid linked follow-up (i.e. it is not `cancelled` and not `no-show`)

Rules:

- The interface is read-only; Lab Manager and Referral Manager MUST NOT mutate appointment state through it
- If no linked appointment exists or the linked appointment is in an invalid state, the interface MUST return an explicit negative result rather than an empty response
- An explicit acknowledged exception recorded in Lab Manager overrides the linkage requirement on the Lab Manager side; Appointment Manager has no obligation to record or validate such exceptions
- `LabRequired` and `LabCaseReference` fields on the Appointment record (§3.1) are owned by Lab Manager and treated as read-only by Appointment Manager

14. Operational Read Feed – Today's Appointments

Appointment Manager MUST publish a today-only, rota-context-filtered appointment feed for downstream operational surfaces (e.g. Surgery & Decon Day View).

Feed characteristics:

- Scoped to the current calendar day (midnight-to-midnight, local practice time zone)
- Filtered by the consuming surface's rota assignment context (e.g. surgery room, nurse rota slot) as supplied in the request
- Updated in real time as appointment states change throughout the day
- Read-only; consumers MUST NOT use this feed to mutate appointment state

Feed record MUST include, at minimum:

Field	Description
<code>appointmentId</code>	Unique identifier
<code>patientId</code>	Patient identifier
<code>patientDisplayName</code>	Display name for operational use
<code>practitionerId</code>	Assigned practitioner
<code>surgeryId</code>	Surgery room
<code>appointmentTypeId</code>	Appointment type
<code>slotStart</code>	UTC timestamp of slot start
<code>slotEnd</code>	UTC timestamp of slot end

<code>lifecycleState</code>	Current state (<code>confirmed</code> , <code>arrived</code> , <code>completed</code> , <code>no-show</code> , <code>cancelled</code>)
<code>accessibilityFlags</code>	Active accessibility or additional needs flags relevant to the session
<code>toothNumbers</code>	Array of tooth identifiers (nullable; populated where clinically applicable)
<code>labRequired</code>	Boolean — whether a lab case is associated with this appointment
<code>labCaseReference</code>	Lab Manager case reference identifier (nullable)

Consumers are responsible for applying any further display-level filtering appropriate to their context. Appointment Manager MUST NOT embed presentation logic in the feed.

15. Read-Only Data Exposure Interface

Appointment Manager MUST expose a governed read-only data interface for external consumers including Smart Dashboards.

This interface:

- Provides live appointment state including lifecycle status for today's appointments
- Surfaces unconfirmed, cancelled, and no-show appointment signals
- Exposes active waitlist match availability for relevant patients
- Exposes practitioner personal schedule data scoped to the requesting context
- Is read-only in all cases; no external consumer may mutate appointment state through this interface

Governance rules:

- Access MUST be subject to role-based and context-based authorisation
- The interface MUST NOT expose patient or clinical data beyond what is necessary for the consuming surface's stated purpose
- All reads against this interface MUST be logged for audit purposes
- Response payloads MUST reflect the current authoritative state; no caching that introduces stale state beyond a practice-configurable staleness threshold is permitted

16. Task-Trigger Event Contract

When an appointment lifecycle transition creates a downstream task obligation, Appointment Manager MUST emit an `AppointmentTaskTriggerEvent` to Task Manager.

Trigger conditions include (but are not limited to):

- Appointment `confirmed` → triggers any configured pre-appointment task (e.g. patient reminder preparation)
- Appointment `completed` → triggers any configured post-appointment task (e.g. follow-up action, referral)

- Appointment `no-show` → triggers any configured no-show follow-up task
- Appointment `cancelled` → triggers any configured cancellation task (e.g. waitlist notification)

Event payload MUST include:

Field	Description
<code>appointmentId</code>	Unique identifier of the appointment
<code>patientId</code>	Patient identifier
<code>practitionerId</code>	Assigned practitioner at time of trigger
<code>appointmentTypeId</code>	Appointment type
<code>triggerTransition</code>	The lifecycle transition that caused the event (confirmed, completed, no-show, cancelled)
<code>transitionTimestamp</code>	UTC timestamp of the lifecycle transition
<code>taskTypeHint</code>	Optional hint identifying the expected task type, as configured in the Appointment Type
<code>creationSource</code>	MUST be set to <code>Appointments</code> verbatim for consistent audit attribution in Task Manager

Task Manager is responsible for task creation, assignment, and execution. Appointment Manager's responsibility ends at event emission. If Access Manager revokes a user's access while tasks derived from this event are in flight, Task Manager is responsible for reverting task ownership to the appropriate role queue; Appointment Manager has no further obligation after emission.

17. Audit & Compliance

The system MUST log the following as immutable, time-stamped, actor-attributed audit events:

- All appointment lifecycle state transitions, with actor identity and booking source
- All read/write actions on patient-bound appointment data
- Waitlist offer events and outcomes (offer created, engagement locked, confirmed, expired)
- Engagement lock start and expiry timestamps, including whether the engaging user was a direct patient or a delegated Family Profiles guardian
- Reschedule job creation, per-patient status changes, and completion
- Manual intervention events on reschedule jobs or waitlist offers, including actor identity
- AI involvement flags and, for any AI suggestion surface, whether the suggestion was accepted or rejected by a human actor
- Care plan entitlement checks performed at booking time, including the entitlement result, `entitlement_status`, `unlock_date`, and any waiting-period or contribution threshold evaluated
- `cadence_locked` signal evaluations at booking and rescheduling time, including the result

- `BookingEligibility` pre-condition check result at appointment creation time
- Digital Forms gate evaluation at Finish & Send, including form-completion status and whether the gate blocked or permitted the transition
- Deposit policy evaluation and the timestamp of any `DepositCollectionEvent` emitted
- `AppointmentTaskTriggerEvent` emissions, including the lifecycle transition that caused them
- `RotaCancellationEvent` emissions to AI Concierge, including `shortNoticeWindowMinutes` threshold applied
- `AppointmentCompletionEvent` emissions to Aftercare Manager and Recall & Reconnect
- `AppointmentAttendanceEvent` emissions to Rewards Manager
- Performance Dashboards lifecycle event emissions
- AI Guardian signal emissions (booking, completion, cancellation)
- Financial Insights signal emissions (booking creation, modification, cancellation)
- All reads against the read-only data exposure interface (§15)
- `CallThread` hand-off events from Communication Hub that initiated booking flows, including the originating call context reference

All audit data MUST be immutable, exportable, and inspection-ready.

18. Non-Functional Requirements

- **Performance:** Automation state changes (offer lock, expiry, reschedule job updates) MUST NOT degrade diary rendering latency. Availability search responses for Find Next MUST complete within a latency target to be defined and agreed during implementation planning; the target MUST be documented in the deployment configuration guide.
- **Reliability:** Waitlist offers and reschedule jobs MUST be idempotent and race-condition safe. `DepositCollectionEvent` and `AppointmentTaskTriggerEvent` emissions MUST be delivered at-least-once with deduplication handled at the consumer; Appointment Manager MUST NOT silently drop events on transient failure. All outbound lifecycle events (including `AppointmentCompletionEvent`, `AppointmentAttendanceEvent`, and Performance Dashboards signals) are subject to the same at-least-once delivery guarantee.
- **Availability:** As the single system of record for scheduling, Appointment Manager availability directly impacts clinical operations. Availability and graceful degradation targets are to be defined in the platform SLA; at minimum, read-only diary access MUST degrade gracefully when write operations are unavailable.
- **Scalability:** The module MUST support multi-site, multi-practice tenancy. Availability search, feed, and interface responses MUST be scoped strictly to the requesting practice context with no cross-tenant data leakage.
- **Security:** All patient-bound data MUST be encrypted at rest and in transit. Secrets and key management MUST follow platform-wide conventions. The read-only data exposure interface (§15) MUST enforce context-scoped authorisation on every request.
- **Privacy:** Appointment Manager handles patient-identifiable scheduling data. GDPR rights — including subject access, rectification, and erasure — must be honoured in accordance with the platform's data retention and privacy policy. Erasure requests must be evaluated against audit immutability requirements; conflicts must be resolved per the platform's documented policy.
- **Observability:** The module MUST export metrics, structured logs, and distributed traces conforming to platform observability conventions. Key signals include: availability search latency, offer lifecycle transition

rates, reschedule job completion rates, event emission success/failure counts, and read-only interface request rates and error rates. Signal availability guarantees for AI Guardian and other upstream-dependent consumers MUST be documented as part of the module's SLA.

- **Governance:** All automation MUST remain observable through the Offer Monitor (§7.4) and Reschedule Job Monitor (§8.3); no silent state changes are permitted.

19. Integration Summary

- **Rota Manager** — inbound Rota Entry, absence/leave data, and `AbsenceChangeEvent`; sole source of availability truth
- **Communication Hub** — outbound notification trigger events for offer state changes and reschedule actions; inbound `CallThread` hand-off context for booking initiation
- **Integrated Payments** — outbound `DepositCollectionEvent` for deposit collection; payment processing owned by Integrated Payments; deposit policy rules owned by Appointment Manager
- **Task Manager** — outbound `AppointmentTaskTriggerEvent` for lifecycle-driven task creation; task ownership reversion on access revocation owned by Task Manager
- **AI Concierge** — outbound `RotaCancellationEvent` for short-notice cancellation patient recovery
- **Lab Manager** — read-only appointment-linkage query interface for Lab Case advancement gating; `LabRequired` and `LabCaseReference` fields owned by Lab Manager, stored as read-only on Appointment records
- **Surgery & Decon Day View** — outbound today-only operational appointment feed (real-time read) including `toothNumbers`, `labRequired`, and `labCaseReference`
- **Smart Dashboards** — governed read-only data exposure interface for live appointment state
- **Access Manager** — RBAC for all capability access control
- **Audit & Compliance** — immutable event log for all auditable actions
- **Treatment Pipeline Manager** — inbound `BookingEligibility` gate; MUST be `true` before appointment creation is permitted
- **Care Plan Subscriptions** — inbound `entitlement_status`, `unlock_date`, and `cadence_locked` signals consumed at booking time
- **Digital Forms** — inbound form-completion status queried before permitting Finish & Send completion transition
- **Recall & Reconnect** — outbound `AppointmentCompletionEvent`; last-completed-appointment date as input for recall interval calculation
- **Rewards Manager** — outbound `AppointmentAttendanceEvent` on completion for attendance-based accrual and referral reward issuance
- **Aftercare Manager** — outbound `AppointmentCompletionEvent` as authoritative trigger for aftercare workflow initiation
- **Performance Dashboards** — outbound lifecycle events (booking, cancellation, completion, no-show) for real-time metric calculation
- **AI Guardian** — outbound booking, completion, and cancellation signals as upstream appointment signal source
- **Financial Insights** — outbound booking creation, modification, and cancellation signals for production and revenue attribution

- **Referral Manager** — read-only appointment-linkage query interface for referral-linked appointment validation
- **HR & People Manager** — confirmed approved-leave state consumed indirectly via Rota Manager's `AbsenceChangeEvent`; no direct integration

20. Explicit Non-Goals

- Message delivery or notification content management — owned by Communication Hub
- Schedule pattern creation or rota management — owned by Rota Manager
- Task creation, assignment, or execution logic — owned by Task Manager
- Payment settlement, retry logic, or ledger management — owned by Integrated Payments
- Clinical documentation or diagnosis — owned by the PMS
- Autonomous AI booking or cancellation decisions
- Recall interval calculation or next-due date computation — owned by Recall & Reconnect
- Referral workflow management or referrer directory management — owned by Referral Manager
- Aftercare workflow execution — owned by Aftercare Manager
- Rewards accrual logic or referral reward issuance — owned by Rewards Manager
- Performance metric aggregation — owned by Performance Dashboards
- Financial aggregation or revenue attribution computation — owned by Financial Insights
- Leave approval workflow management — owned by HR & People Manager

21. Versioning & Governance

This specification is owned by the Appointment Manager module owner.

Changes to this spec require:

- Review by the MVP module owner
- Impact analysis across all declared related modules
- Version bump (patch for clarifications, minor for capability changes, major for breaking contract changes)

22. Build Contract (Engineering & QA)

22.1 Canonical Data Model

Primary tables and their minimum required fields are derived from §3. Engineering **MUST** implement:

```
appointments (
  appointment_id      UUID PRIMARY KEY,
  patient_id          UUID NOT NULL,
  practitioner_id     UUID NOT NULL,
  surgery_id          UUID NOT NULL,
  appointment_type_id UUID NOT NULL,
  rota_entry_id       UUID NOT NULL,
  lifecycle_state     TEXT NOT NULL,  -- see §3.2
  booking_source      TEXT NOT NULL,  -- staff | patient | system
```

```

created_by          TEXT NOT NULL,
created_at         TIMESTAMPTZ NOT NULL,
audit_trail        JSONB NOT NULL, -- immutable append-only
tooth_numbers      UUID[],          -- nullable; populated where clinically applicable
lab_required       BOOLEAN,         -- nullable; owned by Lab Manager, read-only
lab_case_reference UUID             -- nullable; owned by Lab Manager, read-only
)

reschedule_jobs (
  reschedule_job_id  UUID PRIMARY KEY,
  trigger_absence_source  UUID NOT NULL,
  affected_appointment_ids  UUID[] NOT NULL,
  per_patient_status  JSONB NOT NULL,
  created_at         TIMESTAMPTZ NOT NULL,
  audit_trail        JSONB NOT NULL
)

waitlist_offers (
  offer_id          UUID PRIMARY KEY,
  cancelled_appointment_id  UUID NOT NULL,
  offer_state       TEXT NOT NULL, -- see §7.2
  eligible_patient_ids  UUID[] NOT NULL,
  lock_held_by_patient_id  UUID,
  lock_expires_at    TIMESTAMPTZ,
  created_at         TIMESTAMPTZ NOT NULL,
  audit_trail        JSONB NOT NULL
)

```

Final column types, indexes, and foreign key constraints are to be finalised by Engineering during implementation. The above represents the minimum normative field set derived from §3 and must not be reduced without a spec amendment.

22.2 Core Behaviour Rules

1. Availability **MUST** be derived exclusively from Rota Manager-generated RotaEntry records; any appointment booking attempt against a time range not backed by a valid RotaEntry **MUST** be rejected.
2. RotaEntry records with `ShiftType = Break` **MUST** be treated as unavailable; they **MUST NOT** appear in any availability search, diary view, or self-booking surface.
3. All availability searches **MUST** be bounded by the intersection defined in §4.3; full-diary searches are prohibited.
4. Rota entries **MUST** be consumed without treatment-type filtering; availability is role-based, not treatment-based.
5. An appointment **MUST NOT** transition from `completed`, `no-show`, or `cancelled` to any earlier state.
6. Appointment creation **MUST** be rejected if Treatment Pipeline Manager's `BookingEligibility` field is not `true` at the time of the booking request.
7. A `DepositCollectionEvent` **MUST** be emitted to Integrated Payments on every booking confirmation where the Appointment Type carries a deposit policy; Appointment Manager **MUST NOT** process or retry payment operations.
8. A `RotaCancellationEvent` **MUST** be emitted to AI Concierge on every cancellation falling within the configured short-notice window; the event **MUST** include all fields specified in §10.1.
9. An `AppointmentTaskTriggerEvent` **MUST** be emitted to Task Manager on every qualifying lifecycle transition (§16); `creationSource` **MUST** be set to `Appointments` verbatim.

10. The appointment-linkage interface (§13) MUST return an explicit negative result when no linked appointment exists or the linked appointment is in `cancelled` or `no-show` state; empty responses are prohibited.
11. Waitlist offer engagement lock MUST be enforced atomically; only one patient may hold an engagement lock on a given offer at any point in time.
12. Only one confirmed booking may result from a single Waitlist Offer; subsequent engagement attempts after `confirmed` state MUST be rejected.
13. Delegated users (Family Profiles guardians) MUST be held to the same engagement lock window as direct patients; no extended window is granted.
14. All audit events listed in §17 MUST be written before the corresponding action is considered complete; audit write failures MUST cause the action to fail.
15. The read-only data exposure interface (§15) and the today-only operational feed (§14) MUST reject all write attempts with an explicit error; no consumer may mutate appointment state through these interfaces.
16. Care plan entitlement validation at booking time MUST evaluate `entitlement_status`, `unlock_date`, waiting-period, and minimum-contribution thresholds and record the result in the audit trail before confirmation proceeds.
17. `cadence_locked` signals from Care Plan Subscriptions MUST be evaluated at appointment creation and rescheduling time; booking within a locked interval MUST be rejected with an explicit reason.
18. AI Concierge recovery bookings routed back through Appointment Manager MUST be subject to all standard booking constraints and MUST reference the original `appointmentId` in the replacement booking's audit record.
19. The `in_progress` → `completed` transition via Surgery Tablet Finish & Send MUST be blocked if Digital Forms reports mandatory unsigned forms remain; the attempt MUST be rejected with an explicit reason and logged in the audit trail.
20. Absence-driven rescheduling MUST be triggered only by a confirmed `AbsenceChangeEvent` from Rota Manager; draft, submitted, or unapproved leave states MUST NOT trigger rescheduling logic.
21. `AppointmentCompletionEvent` MUST be emitted to Aftercare Manager and Recall & Reconnect on every transition to `completed`.
22. `AppointmentAttendanceEvent` MUST be emitted to Rewards Manager on every transition to `completed`.
23. Performance Dashboards lifecycle events MUST be emitted on every `confirmed`, `completed`, `no-show`, and `cancelled` transition with all fields specified in §11.3.

22.3 Configuration Surfaces

- **Practice-level settings (Admin Control Plane):** Short-notice cancellation window (minutes), engagement lock duration (minutes and permitted range), waitlist broadcast eligibility rules, deposit policy rules, substitution rules per Appointment Type, staleness threshold for read-only interface
- **Per-Appointment-Type overrides (this module):** Task type hints, deposit policy assignment, clinician substitution permission, pre/post-appointment task configuration
- **Per-user preferences (Access Manager):** Diary lens preference (practitioner-first vs surgery-first); other user-level preferences as defined during implementation

22.4 Filtering & Views

The UI MUST support the following standard filters and views:

- Diary views: Day, Week, Month; Practitioner-first and Surgery-first lenses
- Day view: Real-time current-time rendering layer (§5.4)
- Find Next: Persistent modal overlay with governed availability results (§6)
- Offer Monitor: One row per cancelled slot; filterable by offer state (§7.4)
- Reschedule Job Monitor: Job-level and per-patient status views with manual takeover controls (§8.3)

22.5 Module Extension Map

The following extension points are intended to allow future capability growth without breaking this contract:

- Appointment Type configuration is the primary extension surface for adding new scheduling constraints, task configurations, or deposit policies without schema changes
- The `AppointmentTaskTriggerEvent` payload includes `taskTypeHint` to allow new task categories to be configured without changes to the event contract
- The read-only data exposure interface (§15) is designed for additional consumer surfaces to be onboarded by adding authorised consumer registrations, without changes to the interface contract
- The outbound event set (§11.2) supports additional downstream consumers being registered against existing event types (e.g. `AppointmentCompletionEvent`) without changes to event shape
- Future additions to the outbound event set (§11.2) require a minor version bump to this spec and impact analysis across declared related modules

22.6 Acceptance Criteria

The build of Appointment Manager is complete when:

- [] All canonical objects (Appointment, Reschedule Job, Waitlist Offer) can be created, read, and updated through the API with all minimum required fields enforced
- [] State machine transitions enforce all rules in §3.2; prohibited transitions are rejected with explicit errors
- [] Digital Forms gate is enforced at Finish & Send; mandatory unsigned forms block `in_progress` → `completed` transition with an explicit rejection
- [] Availability search is bounded per §4.3; full-diary searches are structurally impossible
- [] Rota entries are consumed without treatment-type filtering
- [] Break segment enforcement (§4.2) is verified: no `ShiftType = Break` `RotaEntry` is ever surfaced as bookable
- [] `BookingEligibility` pre-condition is enforced: booking creation is rejected when `BookingEligibility ≠ true`
- [] Care plan `cadence_locked` signal is enforced: booking within a locked interval is rejected with an explicit reason
- [] All integrations in §11 are wired and verified end-to-end
- [] `DepositCollectionEvent`, `RotaCancellationEvent`, and `AppointmentTaskTriggerEvent` payloads match contracts in §10.1, §12, and §16 respectively
- [] `AppointmentCompletionEvent` is emitted to Aftercare Manager and Recall & Reconnect on every `completed` transition
- [] `AppointmentAttendanceEvent` is emitted to Rewards Manager on every `completed` transition
- [] Performance Dashboards lifecycle events are emitted for all qualifying transitions with correct payload (§11.3)

- [] AI Guardian and Financial Insights outbound signals are emitted and verified
- [] Lab Manager appointment-linkage interface (§13) returns explicit negative results for invalid or absent links
- [] Referral Manager appointment-linkage query is supported and returns explicit negative results correctly
- [] Today's appointment feed (§14) and read-only data interface (§15) reject write attempts; `toothNumbers`, `labRequired`, and `labCaseReference` fields are present in feed records
- [] AI boundaries in §10 are enforced; negative tests confirm AI cannot auto-book or auto-cancel
- [] Audit log captures every event in §17; audit write failure causes action failure
- [] Delegated user (Family Profiles) engagement lock window enforcement is verified
- [] Absence-driven rescheduling is triggered only by confirmed `AbsenceChangeEvent` from Rota Manager; draft or submitted leave states do not trigger rescheduling
- [] CallThread hand-off from Communication Hub correctly initiates a governed booking flow
- [] Access control is enforced per §9 for all capability surfaces
- [] All non-functional requirements in §18 are met and verified
- [] Offer Monitor (§7.4) and Reschedule Job Monitor (§8.3) correctly reflect real-time automation state

23. Open Questions

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

1. **Short-notice cancellation window default:** The default value for the short-notice cancellation threshold (§10.1) is deferred to implementation; it must be defined and documented in the deployment configuration guide before go-live.
2. **Engagement lock window range:** The permitted minimum and maximum engagement lock duration (§7.3) must be defined and agreed during implementation planning.
3. **Performance latency targets:** Specific latency and throughput targets for availability search, diary rendering, and event emission (§18) are not yet defined; these must be agreed and documented before the build contract is finalised.
4. **Availability degradation SLA:** The graceful degradation behaviour and availability target for Appointment Manager as the single system of record for scheduling (§18) must be defined in the platform SLA.
5. **RBAC role reconciliation:** The role names in §9 are indicative and must be reconciled against the authoritative Access Manager role registry before the spec is promoted.
6. **MFA requirements:** Whether MFA is required for sensitive operations (bulk cancellation, absence-driven reschedule job initiation) must be confirmed with the Access Manager specification.
7. **GDPR erasure vs audit immutability conflict:** The platform's documented policy for resolving conflicts between GDPR erasure requests and immutable audit log requirements (§18) must be referenced or defined before go-live.
8. **Referral Manager linkage scope:** The full extent of Referral Manager's query requirements against the appointment-linkage interface (§13) — including whether inbound referrals create appointment stubs or merely link to independently created appointments — must be confirmed with the Referral Manager module owner before the build contract is finalised.
9. **AppointmentCompletionEvent schema:** The canonical payload for `AppointmentCompletionEvent` (consumed by Aftercare Manager, Recall & Reconnect, and potentially others) must be defined and agreed across consuming modules before implementation.

10. **Financial Insights signal schema:** The minimum field set for booking creation, modification, and cancellation signals to Financial Insights must be agreed with the Financial Insights module owner during implementation planning.