

Rota Manager

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

Rota Manager – Technical Specification

1. Module Purpose & Scope (Authoritative)

Rota Manager is the single system of record for staff availability, coverage, and operational readiness across Primoro. It defines when staff are available to work, in which role, and at which site, and provides Appointment Manager with accurate availability inputs. The rota is not a diary — it is the authoritative availability input layer.

It governs:

- Defining availability patterns that generate date-specific rota entries
- Mapping staff to roles, sites, and time windows
- Enforcing shift protection (no rota = no availability)

It explicitly does not:

- Determine appointment suitability, treatment types, durations, or zoning (Appointment Manager)
- Schedule or execute tasks (Task Manager)
- Communicate with patients (Communication Hub)
- Manage payroll, time sheets, or HR records (HR & People Manager)
- Own clinical documentation (PMS)

2. Ownership & Responsibilities

2.1 Rota Manager IS Responsible For

- Defining Schedule Patterns per staff member
- Mapping staff to roles, sites, and time windows
- Generating date-specific rota entries from patterns
- Managing shifts across locations and sites
- Enforcing shift protection (no rota = no availability)
- Modelling breaks and protected non-working time
- Integrating approved absences that override patterns
- Providing staff visibility via Staff App Mode and read-only feeds
- Maintaining a full audit trail of all schedule and rota changes
- Emitting structured absence-change events on all absence state transitions

2.2 Rota Manager IS NOT Responsible For

- Defining treatment types or appointment reasons (Appointment Manager)
- Applying booking rules, zones, or durations (Appointment Manager)
- Scheduling or executing tasks (Task Manager)
- Communicating with patients (Communication Hub)
- Payroll, time sheets, or HR records (HR & People Manager)
- Clinical documentation (PMS)
- Determining which appointments are affected by an absence or executing reschedule logic (Appointment Manager)
- Enforcing HR approval logic (HR & People Manager)

3. Core Objects (Normative)

3.1 Schedule Pattern (Canonical Artefact)

A Schedule Pattern is a governed digital artefact representing a recurring availability definition for a staff member.

Minimum required fields:

- PatternId
- StaffId
- RoleId
- LocationId
- RecurrenceType (Weekly / Fortnightly)
- StartDate
- Days[] — DayOfWeek, StartTime, EndTime
- Breaks[] — StartTime, EndTime, Label (zero or more per day)
- Active
- AuditLog (immutable)

Patterns are the authoritative source of future availability.

3.2 Rota Entry (Derived Object)

A Rota Entry is a date-specific instance generated from a Schedule Pattern.

Minimum required fields:

- RotaEntryId
- PatternId (nullable if override)
- StaffId
- RoleId
- LocationId
- StartDateTime / EndDateTime
- EntryType (Generated / Override)
- ShiftType (Working / Break / WorkPackage)

- AuditLog (immutable)

Field contract for downstream consumers. `RotaEntryId` is the stable, globally unique reference that downstream consumers MUST use when citing a specific rota slot. Consumers MUST NOT derive availability by reading Schedule Pattern records directly; only materialised Rota Entries are authoritative for a given date. `ShiftType` carries the following semantics that all consumers depend on:

- `Working` — the staff member is present and bookable within the window
- `Break` — protected non-working time; non-bookable and invisible to Appointment Manager as an available slot
- `WorkPackage` — a task-manager-surfaced work block within a working shift; treated as non-bookable by Appointment Manager but visible to Task Manager

Appointment Manager MUST use `RotaEntryId` as the foreign key when recording which rota slot underpins a booked appointment. Rota Manager MUST NOT remove or reassign a `RotaEntryId` once emitted; historical identifiers are immutable.

3.3 Rota Entry State Machine (Authoritative)

States:

- `Generated` — automatically created from a pattern
- `Override` — manually overridden instance

Rules:

- Rota entries are read-only by default; only manual overrides alter a specific entry
- Manual overrides MUST NOT alter the underlying pattern
- State transitions are auditable and time-stamped
- Historical rota entries MUST NOT be mutated by pattern changes
- Pattern changes affect future availability only

3.4 Absence Change Event (Canonical Artefact)

An `AbsenceChangeEvent` is emitted by Rota Manager whenever an absence is confirmed, amended, or cancelled.

Required fields:

- `EventId`
- `StaffId`
- `StartDate / EndDate`
- `LocationId[]`
- `RoleId[]`
- `EventType`: `AbsenceConfirmed` / `AbsenceAmended` / `AbsenceCancelled`
- `Timestamp`
- `Source`: `ManualOverride` / `HRApprovedFeed`

A confirmed absence MUST emit an `AbsenceChangeEvent` before downstream consumers act.

3.5 Out-of-Hours Window (Derived Signal)

Rota Manager is the authoritative source for determining when a staff member's working period begins and ends on any given day. This information is used by downstream consumers — including Staff App Mode (for quiet hours enforcement) and AI Concierge (for out-of-hours and overflow window detection) — to determine when practice staff are scheduled and when out-of-hours periods are active.

A working day is considered active for a given staff member when at least one `ShiftType: Working` Rota Entry exists for that `StaffId` and date at the relevant `LocationId`. Where no such entry exists, the full day is treated as outside working hours by all consumers of this signal.

Rota Manager MUST NOT itself make decisions about out-of-hours routing or AI activation; those decisions belong to their respective modules. Rota Manager's obligation is to expose accurate, up-to-date shift start and end times so that consumers can derive these windows reliably.

4. Scheduling Capabilities

4.1 Shift Manager (Pattern Control Layer)

Rota Manager MUST expose a Shift Manager interface for management.

The module MUST:

- Provide the Shift Manager as the system of authority for schedule patterns
- Support creation, editing, activation, and deactivation of patterns via the Shift Manager
- Support weekly and fortnightly recurrence logic
- Support multiple breaks per day within a pattern
- Ensure pattern changes affect future availability only and never mutate historical rota entries

The module MUST NOT:

- Allow any consumer other than the Shift Manager to write to or modify rota or pattern state

4.2 Breaks and Non-Working Time (Authoritative)

Breaks are first-class segments within a shift pattern.

Required:

- Breaks reduce working availability
- Breaks are non-bookable
- Breaks are generated into the rota as protected time

Optional (configurable):

- Break type label: unpaid lunch break, paid break, or protected non-working block

4.3 Multi-Site Scheduling (Authoritative)

The module MUST:

- Support multi-site group practices
- Allow a staff member to have multiple patterns, each scoped to a specific site
- Require explicit pattern definition for cross-site availability

The module MUST NOT:

- Apply automatic cross-site scheduling

4.4 Absence and Exception Handling (Authoritative)

Approved absences override generated rota entries and temporarily override underlying schedule patterns.

The module MUST:

- Remove availability immediately on absence confirmation
- Preserve base patterns — absence MUST NOT mutate them
- Emit a structured `AbsenceChangeEvent` (see Section 3.4) on every absence state transition
- Surface absence impact to downstream consumers (via Appointment Manager as primary consumer)

The module MUST NOT:

- Determine which appointments are affected by an absence
- Execute reschedule logic (Appointment Manager)

4.5 Attendance Signals (Advisory)

Rota Manager MAY record attendance signals derived from system or mobile activity.

Signals include:

- Arrived on site
- No arrival detected
- Likely late

The module MUST NOT:

- Use attendance signals for payroll purposes
- Automatically modify rota entries based on attendance signals

All attendance signals are advisory only.

5. Delivery Surfaces & Access (Authoritative)

5.1 Web Portal

Rota Manager is the system of record for staff availability in the staff web portal. The Shift Manager interface for creating, editing, activating, and deactivating schedule patterns is available via the web portal.

5.2 Tablet App

Surgery & Decon Day View consumes generated rota entries as a read-only feed from Rota Manager to filter tablet surfaces by rota-assigned surgery and nurse context and to present accurate room and staff assignment context at the point of care.

5.3 Patient Mobile App

(no content captured in original — needs definition)

5.4 Engagement Signals

Rota Manager provides staff visibility via Staff App Mode and read-only availability feeds to downstream consumers. Attendance signals (arrived on site, no arrival detected, likely late) are emitted as advisory signals for staff visibility.

6. Integration Contracts

6.1 Inbound (this module consumes from)

From module	What	Contract
HR & People Manager	Approved absence feed	Async / HR-approved feed

6.2 Outbound (this module emits to)

To module	What	Contract
Appointment Manager	Generated rota availability; AbsenceChangeEvent on all absence state transitions	Read-only feed; event
Task Manager	Role-scoped availability feed; WorkPackage shift type overlay within rota entries	Read-only queryable feed
Communication Hub	Staff notification routing only	Read-only feed
Staff App Mode	Shift start and end times per staff member (quiet hours / notification suppression)	Read-only feed (behavioural dependency)
HR & People Manager	Availability check, coverage query, and conflict detection queries for leave approval workflow	Read-only query interface
Surgery & Decon Day View	All active rota entries per LocationId and date (StaffId, RoleId, ShiftType, room/surgery context)	Read-only feed
AI Concierge	Shift start/end times and active-pattern presence per LocationId and date (out-of-hours window detection)	Read-only query interface

Financial Insights	Active rota entry presence and absence change events per StaffId and LocationId (advisory; for provider statement filtering and site-level accuracy)	Read-only feed (advisory)
--------------------	--	---------------------------

6.3 Detailed Integration Contracts

Appointment Manager

Appointment Manager consumes generated rota availability to determine bookable windows. It MUST be the primary consumer of `AbsenceChangeEvents` and is responsible for initiating any reschedule jobs arising from a practitioner absence. Rota Manager's obligation ends at reliable, timely event emission.

Appointment Manager MUST treat materialised Rota Entries as the sole source of availability truth and MUST NOT infer availability from Schedule Patterns directly. When recording a booked appointment, Appointment Manager MUST reference the `RotaEntryId` of the underpinning slot. Rota Manager guarantees that `RotaEntryId` values are stable and immutable once emitted; Appointment Manager MAY cache them safely for the lifetime of the booking.

Task Manager — Availability Feed Contract

Rota Manager MUST expose a queryable availability feed that Task Manager can consume. This feed MUST support:

- Query by StaffId and date range
- Query by RoleId and LocationId for a given time window
- Response payload including: ShiftType, StartDateTime, EndDateTime, RoleId, LocationId, and Active status

Freshness requirement. The availability feed MUST reflect the current materialised state of rota entries. Where a pattern change or absence override has been committed, the feed MUST return the updated state within a bounded propagation window; the target latency for feed updates relative to a write commit is subject to the performance targets defined in Section 14 (and open in Section 15). Task Manager MUST NOT cache availability data beyond this propagation window for claiming and assignment decisions.

Task Manager uses this feed to determine which tasks and work packages are visible to a given staff member at any point in time, filtering out assignments that fall outside an active working shift. The Work Package overlay (whereby work packages are surfaced as `ShiftType: WorkPackage` entries within rota patterns) is a distinct mechanism from task visibility filtering; both contracts operate concurrently.

Staff App Mode — Quiet Hours Contract

Rota Manager MUST make each staff member's shift start and end times available to Staff App Mode. Staff App Mode consumes this data to:

- Determine the notification delivery window for that staff member
- Suppress non-urgent notifications outside working hours (quiet hours enforcement)
- Resume notification delivery at shift start

The feed MUST include, per staff member per calendar day: `StaffId`, `LocationId`, `ShiftType`, `StartDateTime`, `EndDateTime`, and `Active` status. Only entries with `ShiftType: Working` define the boundary of the notification delivery window; `Break` and `WorkPackage` entries do not extend the window.

Staff App Mode MUST NOT apply quiet hours logic independently of Rota Manager shift data. Where no rota entry exists for a given day, Staff App Mode SHOULD treat the full day as outside working hours and suppress accordingly, unless overridden by a user preference.

HR & People Manager — Pre-Approval Availability Query Contract

Rota Manager MUST expose a read interface supporting the following queries, invoked by HR & People Manager as part of its leave approval workflow:

- **Availability check** — given a StaffId and date range, return all generated rota entries and any existing absence overrides for that period
- **Coverage query** — given a RoleId, LocationId, and date range, return all staff with active patterns covering that window
- **Conflict detection** — return any rota entries that would be removed or blocked by a proposed absence, including role-based dependencies and flagged protected duties

Conflict detection MUST operate against date-resolved Rota Entry records, not raw Schedule Patterns.

This ensures that any manual overrides, existing absence blocks, or WorkPackage overlays already in effect for the proposed leave period are visible to HR & People Manager at the point of conflict evaluation. HR & People Manager MUST NOT attempt to derive conflict data by reading Schedule Pattern records directly.

Rota Manager provides raw availability and conflict data only. HR & People Manager is responsible for applying approval rules, risk thresholds, and organisational policy. Rota Manager MUST NOT enforce HR approval logic. HR & People Manager MAY also consume AbsenceChangeEvents to reconcile leave records with rota state.

Surgery & Decon Day View — Day-of Rota Feed Contract

Rota Manager MUST make the following available per calendar day:

- All active rota entries for a given LocationId and date
- StaffId, RoleId, and ShiftType for each entry
- Assigned surgery or room context where captured within the rota entry

Filtering semantics. The feed MUST return only entries with `EntryType: Generated` or `EntryType: Override` that have not been superseded by an absence override for the same StaffId, RoleId, and date. Surgery & Decon Day View relies on this feed as the sole gate for appointment visibility on the tablet surface: an appointment MUST only be surfaced by Day View if a valid, non-absent Rota Entry exists for the relevant practitioner and nurse context on that date and at that LocationId. Rota Manager MUST therefore ensure that absence overrides are reflected in the day-of feed within the same propagation window as the broader availability feed.

Day View MUST NOT write to or modify rota entries via this feed.

AI Concierge — Out-of-Hours Window Contract

Rota Manager MUST expose a read interface that AI Concierge can query to determine whether practice staff are scheduled on a given date and at a given LocationId, and when their working period begins and ends. AI Concierge uses this to:

- Determine when out-of-hours periods are active (no `ShiftType: Working` entry present for the location and date)
- Surface the next scheduled working window in the queue view when the practice is in an out-of-hours or overflow state

The response **MUST** include: `LocationId`, `Date`, whether at least one `ShiftType: Working` entry is present, and the earliest `StartDateTime` and latest `EndDateTime` across all `Working` entries for that location and date. Where no working entry exists, the response **MUST** explicitly indicate that the location has no scheduled staff for that day rather than returning an empty payload.

AI Concierge **MUST NOT** use this interface to make routing or escalation decisions autonomously; it is an input signal only. Rota Manager **MUST NOT** implement any AI routing logic.

Financial Insights — Advisory Availability Contract

Rota Manager **MAY** make active rota entry presence and `AbsenceChangeEvents` available to Financial Insights on a read-only, advisory basis. Financial Insights **MAY** consume this data to:

- Filter provider statements by actual scheduled availability (e.g. excluding days with confirmed absence)
- Improve accuracy of site-level performance metrics where a provider's absence would otherwise skew denominator calculations

This contract is advisory: Financial Insights is not a primary consumer of rota data and **MUST NOT** treat Rota Manager as a financial source of record. The availability feed query interface described in the Task Manager contract (filtered by `StaffId` and date range) is sufficient for this purpose; no dedicated Financial Insights endpoint is required. Rota Manager **MUST NOT** implement financial calculation logic.

6.4 PMS Boundary

Primero remains authoritative for availability when configured as primary. Rota Manager governs availability only; clinical documentation remains in the PMS.

7. AI Boundaries (Non-Negotiable)

Module does not embed AI surfaces that act autonomously on rota or pattern state. AI is available as an advisory layer only, surfaced through the staff web portal.

AI **MAY**:

- Explain scheduling options to staff
- Summarise rota activity for human review
- Suggest content from approved libraries (with explicit human approval before commit)

AI **MAY NOT**:

- Auto-decide on policy-bound scheduling actions
- Modify rota or pattern state
- Bypass governance, audit, or access checks
- Make commitments on behalf of the practice
- Replace required clinical judgement

8. Audit & Compliance

The system **MUST** log:

- All state transitions on canonical objects (`Schedule Pattern`, `Rota Entry`, `AbsenceChangeEvent`), with actor and timestamp

- All read/write actions on patient-bound or staff-bound data
- All AI suggestions, including which were accepted or rejected by humans
- All cross-module events emitted or consumed

Audit logs MUST be immutable and exportable for inspection.

The following named audit events are emitted by Rota Manager:

- `pattern.created` — a new Schedule Pattern is saved, with actor, StaffId, RoleId, and LocationId
- `pattern.updated` — an existing Schedule Pattern is modified, with before/after snapshot and actor
- `pattern.activated / pattern.deactivated` — Active flag toggled, with actor and timestamp
- `rota_entry.generated` — a rota entry is generated from a pattern, with PatternId reference
- `rota_entry.overridden` — a manual override is applied to a generated entry, with actor and reason
- `absence_change_event.emitted` — AbsenceChangeEvent fired (type: AbsenceConfirmed / AbsenceAmended / AbsenceCancelled), with Source and downstream acknowledgement status
- `availability_feed.queryed` — a downstream consumer queries the availability feed, with consumer identity and query parameters

9. Access Control

(Access control role definitions are not captured in the original — needs definition by module owner in conjunction with Access Manager.)

References to Access Manager roles. The following capabilities require access control definition:

- Create and edit Schedule Patterns (Shift Manager interface)
- Read rota entries (downstream consumers via read-only feeds)
- Override individual rota entries
- Approve / promote pattern changes
- Delete or deactivate patterns

Access Manager is the authoritative source for role definitions. Rota Manager MUST enforce RBAC decisions supplied by Access Manager and MUST NOT implement its own role store.

Whether MFA is required for sensitive operations (e.g. pattern deactivation, bulk override) is an open question (see Section 15).

10. Integration Summary

- **Appointment Manager** — outbound: generated availability feed and AbsenceChangeEvents; primary consumer of absence signals for reschedule jobs; references `RotaEntryId` as stable foreign key on bookings
- **Task Manager** — outbound: role-scoped availability feed and WorkPackage overlay for task visibility filtering; feed freshness bounded by propagation window
- **Communication Hub** — outbound: staff notification routing only
- **Staff App Mode** — outbound: shift start/end times for quiet hours enforcement (behavioural dependency); `Working` entries define notification delivery window boundaries
- **HR & People Manager** — outbound: availability check, coverage query, and conflict detection against date-resolved Rota Entries for leave approval; inbound: approved absence feed

- **Surgery & Decon Day View** — outbound: read-only daily rota feed for day-of assignment context; filtering semantics gate appointment visibility on tablet surface
- **AI Concierge** — outbound: shift presence and working-window bounds per LocationId and date for out-of-hours detection
- **Financial Insights** — outbound: active rota entry presence and AbsenceChangeEvents (advisory) for provider statement accuracy and site-level metric filtering
- **Access Manager** — RBAC for read/write/approve on patterns and rota entries
- **Audit & Compliance** — immutable event log for all state transitions and cross-module events

11. Explicit Non-Goals

- Creating appointments — owned by Appointment Manager
- Evaluating treatment suitability — owned by Appointment Manager
- Applying booking rules, zones, or durations — owned by Appointment Manager
- Automatic staffing optimisation — not in scope; would require a separate optimisation module
- Executing tasks — owned by Task Manager
- Managing payroll or time sheets — owned by HR & People Manager
- Clinical documentation — owned by PMS
- Implementing financial calculation logic or acting as a financial source of record — owned by Financial Insights and its upstream sources

12. Versioning & Governance

This specification is owned by: *(role — needs definition)*

Changes to this spec require:

- Review by MVP module owner
- Preservation of pattern-first scheduling, role-based availability, separation from Appointment Manager, and full auditability
- Impact analysis across declared related modules (see /propose)
- Version bump (patch / minor / major)

13. Build Contract (Engineering & QA)

13.1 Canonical Data Model

```

schedule_patterns (
  PatternId      UUID PRIMARY KEY,
  StaffId        UUID NOT NULL,
  RoleId         UUID NOT NULL,
  LocationId     UUID NOT NULL,
  RecurrenceType ENUM('Weekly', 'Fortnightly') NOT NULL,
  StartDate      DATE NOT NULL,
  Active         BOOLEAN NOT NULL,

```

```

AuditLog          JSONB NOT NULL -- immutable append-only
)

schedule_pattern_days (
  PatternDayId     UUID PRIMARY KEY,
  PatternId        UUID NOT NULL REFERENCES schedule_patterns(PatternId),
  DayOfWeek        ENUM('Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun') NOT NULL,
  StartTime        TIME NOT NULL,
  EndTime          TIME NOT NULL
)

schedule_pattern_breaks (
  BreakId          UUID PRIMARY KEY,
  PatternDayId     UUID NOT NULL REFERENCES schedule_pattern_days(PatternDayId),
  StartTime        TIME NOT NULL,
  EndTime          TIME NOT NULL,
  Label            TEXT
)

rota_entries (
  RotaEntryId     UUID PRIMARY KEY,
  PatternId        UUID REFERENCES schedule_patterns(PatternId), -- nullable if override
  StaffId         UUID NOT NULL,
  RoleId          UUID NOT NULL,
  LocationId      UUID NOT NULL,
  StartDateTime   TIMESTAMPTZ NOT NULL,
  EndDateTime     TIMESTAMPTZ NOT NULL,
  EntryType       ENUM('Generated', 'Override') NOT NULL,
  ShiftType       ENUM('Working', 'Break', 'WorkPackage') NOT NULL,
  AuditLog        JSONB NOT NULL -- immutable append-only
)

absence_change_events (
  EventId         UUID PRIMARY KEY,
  StaffId         UUID NOT NULL,
  StartDate       DATE NOT NULL,
  EndDate         DATE NOT NULL,
  LocationIds     UUID[] NOT NULL,
  RoleIds         UUID[] NOT NULL,
  EventType       ENUM('AbsenceConfirmed', 'AbsenceAmended', 'AbsenceCancelled') NOT NULL,
  Timestamp       TIMESTAMPTZ NOT NULL,
  Source          ENUM('ManualOverride', 'HRApprovedFeed') NOT NULL
)

```

13.2 Core Behaviour Rules

1. No pattern → no availability. A staff member with no active Schedule Pattern has no bookable rota entries.
2. Pattern changes generate future rota entries only; historical rota entries **MUST NOT** be mutated.
3. Breaks reduce availability; break segments are non-bookable and generated into the rota as protected time.
4. Manual overrides to rota entries **MUST NOT** alter the underlying pattern.
5. A confirmed absence **MUST** emit an `AbsenceChangeEvent` before downstream consumers act.
6. `AbsenceChangeEvents` **MUST** be emitted on every absence state transition: `AbsenceConfirmed`, `AbsenceAmended`, and `AbsenceCancelled`.

7. AI MUST NOT modify rota or pattern state under any circumstances.
8. All availability feeds exposed to Task Manager, HR & People Manager, Staff App Mode, Surgery & Decon Day View, AI Concierge, and Financial Insights are read-only; no consumer other than the Shift Manager may alter rota or pattern state.
9. Attendance signals are advisory only and MUST NOT automatically modify rota entries.
10. Cross-site availability requires an explicit pattern per site; no automatic cross-site scheduling is permitted.
11. `RotaEntryId` values are immutable once emitted; downstream consumers (including Appointment Manager) MAY safely use them as stable foreign key references.
12. Conflict detection queries from HR & People Manager MUST be resolved against materialised Rota Entry records, not raw Schedule Pattern records.
13. The day-of rota feed supplied to Surgery & Decon Day View MUST exclude entries superseded by an absence override for the same StaffId, RoleId, and date.

13.3 Configuration Surfaces

- **Practice-level settings (Admin Control Panel):** multi-site configuration, recurrence type defaults, break type labels
- **Per-user preferences (Access Manager):** notification and quiet hours overrides
- **Per-pattern overrides (Shift Manager):** site-scoped pattern activation, break segments, fortnightly rotation configuration

13.4 Filtering & Views

Standard filters and saved views the UI must support:

- Filter rota entries by StaffId, RoleId, LocationId, and date range
- Filter availability feed by ShiftType
- Day view filtered by LocationId (consumed by Surgery & Decon Day View)
- Coverage view by RoleId and date range (consumed by HR & People Manager)
- Pattern management view filtered by Active status and recurrence type

13.5 Module Extension Map

- Additional recurrence types (e.g. monthly, custom cycles) MAY be added by extending `RecurrenceType` without altering existing pattern or rota entry contracts
- Additional ShiftTypes MAY be added as an enum extension without breaking existing consumers
- Additional absence event types MAY be added to `EventType` as the HR integration matures
- The availability query interface MAY be extended with new query parameters without breaking existing consumers provided backwards compatibility is maintained

13.6 Acceptance Criteria

The build of Rota Manager is complete when:

- [] All canonical objects (Schedule Pattern, Rota Entry, AbsenceChangeEvent) can be created, read, and updated through the API
- [] State machine transitions enforce all rules in Section 3

- [] Dentally-equivalent schedule editing behaviour is achieved in the Shift Manager interface
- [] Breaks correctly block availability and are generated into rota entries as protected time
- [] Patterns generate rota entries predictably; no treatment logic is present in the rota
- [] All integrations in Section 6 are wired (Appointment Manager, Task Manager, Communication Hub, Staff App Mode, HR & People Manager, Surgery & Decon Day View, AI Concierge, Financial Insights)
- [] AbsenceChangeEvent are emitted reliably on all absence state transitions
- [] Availability feed is queryable by Task Manager, HR & People Manager, Staff App Mode, Surgery & Decon Day View, AI Concierge, and Financial Insights under defined read-only contracts
- [] `RotaEntryId` is stable and immutable once emitted; negative test confirms no reassignment on pattern change
- [] Day-of rota feed correctly excludes absence-superseded entries (negative test: confirmed absence suppresses Day View slot)
- [] AI boundaries in Section 7 are enforced (negative tests pass)
- [] Audit log captures every event in Section 8
- [] Access control is enforced per Section 9
- [] All non-functional requirements in Section 14 are met

14. Non-Functional Requirements

- **Performance:** *(target latency and throughput thresholds under peak multi-site load — needs definition; see Section 15)*
- **Reliability:** *(availability target and graceful degradation behaviour — needs definition)*
- **Scalability:** Must support multi-site, multi-tenant group practices. A single staff member may hold patterns across multiple sites simultaneously. The availability query feed MUST remain responsive under concurrent queries from multiple downstream consumers (Appointment Manager, Task Manager, HR & People Manager, Surgery & Decon Day View, AI Concierge, Financial Insights) without degrading write path latency for pattern management.
- **Security:** *(encryption at rest and in transit, key management, secrets handling — needs definition)*
- **Privacy:** *(GDPR rights this module honours, data retention policy for rota entries and absence change events — needs definition; see Section 15)*
- **Observability:** *(metrics, structured logs, and distributed traces this module exports — needs definition)* At minimum, the module SHOULD emit metrics for: availability feed query latency, AbsenceChangeEvent emission success/failure rate, rota entry generation lag relative to pattern activation, and audit log write throughput.

15. Open Questions

1. **Access control model:** Which roles may create, edit, activate, and deactivate Schedule Patterns? Is MFA required for pattern changes or bulk overrides?
2. **Pattern version history:** Should historical pattern versions be retained as discrete records for audit purposes, or is the append-only AuditLog field on `schedule_patterns` sufficient to meet compliance requirements?

3. **Attendance signal ingestion:** Which system or mobile signals feed into advisory attendance detection, and what is the ingestion contract for those signals?
4. **Performance targets:** What are the acceptable latency thresholds for the availability query feed under peak multi-site load across all concurrent downstream consumers?
5. **Data retention:** What is the retention policy for historical rota entries and absence change events, and which GDPR rights (erasure, portability) does this module need to honour for staff-bound records?
6. **Spec ownership:** Which named role owns this specification for governance and change-approval purposes?
7. **Feed propagation window:** What is the maximum acceptable propagation delay between a pattern change or absence override commit and the updated state being visible in the availability feed, day-of rota feed, and out-of-hours window query? This threshold governs Task Manager claiming, Surgery & Decon Day View slot gating, and AI Concierge window detection simultaneously.