

— ACT III · APPLIED FHIR

A telemedicine visit is four FHIR resources — plus two layers FHIR does not model.

Underneath the booking is a chain of four scheduling resources. Bolted on by convention: a video room, and a searchable index of the patient's clinical documents.

● Scheduling quartet

● Calendar flow

● Video tie-in

● Clinical RAG

— DIAGRAM · THE SCHEDULING QUARTET

Four resources, one reference chain.

The reference chain

Read it top to bottom — each resource points at the one above.

↑ `Schedule.actor`

↑ `Slot.schedule`

SL

Slot

one window — free, busy, with start and end

↑ `Appointment.slot`

AP

Appointment

consumes Slots, names attendees via `participant.actor`

— WHY PRACTITIONERROLE HEADS THE CHAIN

PractitionerRole binds the clinician to an organization, a location, and a specialty. The calendar is for "Dr Okonkwo as a cardiologist at the Harbor Clinic" — not the bare Practitioner.

— PARTICIPANT.ACTOR RE-ASSERTS THE BOOKING

On the Appointment itself, `participant.actor` names both the patient and the practitioner. The booking is self-describing — you never have to walk back up the chain to know who is in the room.

— WALKTHROUGH · APPOINTMENT.STATUS

The happy path, and the three branches off it.

Happy path

Ten codes in the value set — these five are the spine.



3 arrived / checked-in
present in clinic or joined the room



4 fulfilled
the encounter is complete



— OFF THE HAPPY PATH

● **cancelled**

● **noshow**

● **waitlist**

— TWO CONSTRAINTS TO MEMORIZE

Only proposed or cancelled may omit start and end — anything booked or later must be timed. A cancellation reason applies only when the status is cancelled or noshow.

— PARTICIPANT.STATUS ROLLS UP

Each participant carries accepted, declined, tentative, or needs-action — often mediated by an AppointmentResponse — and those roll up into the overall status.

— CONCEPT · SLOT AVAILABILITY + SCHEDULE.ACTOR

The Slot is the unit of availability.

Each Slot carries a status and a start and end. Availability lives on the Schedule's actor — that is how a free Slot ties back to a specific PractitionerRole.

— SLOT.STATUS VALUE SET

● free

● busy

● busy-unavailable

● busy-tentative

● entered-in-error

— FIND AN OPEN SLOT

GET Slot?

**schedule.actor=PractitionerRole/123
&status=free&start=ge2026-05-25**

Some servers expose a dedicated find operation for availability — but where that is not available, plain Slot search by status free is the dependable baseline. Confirm which your HAPI deployment supports rather than assume.

— WALKTHROUGH · CALENDAR FLOW — CONCEPTUAL

Discover, book, busy-out, launch.

The shape of the Dudoxx calendar flow

Conceptual — your bindings confirm the details.



3

Busy-out

flip the chosen Slot to busy



4

Launch

the booked Appointment anchors the consult

— THE BFF OWNS THE UNIT OF WORK

The backend performs slot-search-then-book as one server-side unit — it never trusts the browser to pick a slot that might already be taken.

— ATOMIC — CALLBACK TO CHAPTER 05

Steps two and three happen together in a transaction Bundle: book the Appointment and busy-out the Slot atomically, so a race can never leave a booking against a slot someone else grabbed.

— CONCEPT · VIDEO TIE-IN — THE DESIGN SPACE

FHIR has no video-room field. Two ways teams wire it.

Neither LiveKit nor Jitsi is modelled in FHIR — the integration is convention, not a core element. Present both; assert neither.

— OPTION A · DETERMINISTIC ROOM NAME

Derive the room identity from the Appointment logical id — stable and idempotent — and store the join link in Appointment.supportingInformation or a linked endpoint.

— OPTION B · BFF-SIDE MAPPING

Keep a backend-side mapping keyed by the appointment id and store nothing extra on the resource itself.

— THE SECURITY LINE — CALLBACK TO CHAPTER 13

LiveKit mints a signed access token, and Jitsi exposes a room URL with an optional token — always server-side, never from the browser. Access is gated by status: join only a booked appointment, in its window, once marked arrived or checked-in. On disconnect, flip to fulfilled. The exact token schema and mapping field are yours to confirm.

— DIAGRAM · RAG OVER CLINICAL DOCUMENTS

DocumentReference is the index entry — the bytes live in Binary.

Ingest rail

Enumerate, resolve, fetch, then decode by content type.

↓ `content.attachment.url`

↓ extract by `contentType`



Extract → chunk

PDF extraction, plain passthrough, or OCR for scans

↓ embed



Embed → retrieve

vectorize chunks into the index

— THE ATTACHMENT CARRIES THE PAYLOAD

Each `content.attachment` is an Attachment datatype: an `attachment.url` pointing at a Binary or external store, or inline base-64 data — and a `contentType`, say `application/pdf` or `text/plain`, that tells the pipeline how to decode it.

— EXTRACTION STAYS CONCEPTUAL

The steps are fixed — extract, chunk, embed, retrieve. The concrete PDF and OCR tooling is an implementation choice you bind to your own stack.

— DIALOGUE · GROUNDED CLINICAL Q&A

Ask the documents. Answer with a citation.



Clinical Q&A

• patient-scoped

What did the last cardiology note say about this patient's medication?

What the system does

Callback Chapter 18 — this is the tool a FHIR-over-MCP agent calls.

4

Cite the DocumentReference

so a clinician can verify

— WALKTHROUGH · ONE LOOP, END TO END

The calendar produces the encounter. The encounter feeds the next.

The full patient journey

Scheduling and RAG are two halves of one loop.



3

Fulfilled → note

the clinician's note persists as a DocumentReference



4

Index → grounded Q&A

at the next visit, the answer cites the note we just watched get booked

— THE PAYOFF

The calendar produces the encounter, the encounter produces the document, and the document feeds the retrieval that informs the next encounter.

— TWO FHIR RESOURCES, TWO BOLTED-ON LAYERS

● Appointment

● DocumentReference

● Video room

● RAG index

— CONCEPT · PRODUCTION NOTES

Four habits to carry into production.

— ATOMICITY

Book the Appointment and busy-out the Slot in a single transaction Bundle so a race can never double-book.

— SCOPE DISCIPLINE

RAG retrieval is patient and tenant scoped — partition the index the way you partition the data.

— TOKEN DISCIPLINE

The video-room token is minted server-side only, gated by appointment status, never handed out from the browser.

— CITATION DISCIPLINE

Every generated answer carries its source DocumentReference so a clinician can open the original and verify.

● **Confirm against your deployment:**

● **\$find vs Slot?status=free**

● **room-mapping field**

● **extraction stack**

— RECAP · TRY IT, THEN THE BRIDGE

Book a visit, index its note — then pay the audit bill.

— TRY IT · CALENDAR

Search free Slots for a PractitionerRole, POST a booked Appointment in a transaction Bundle that also flips the chosen Slot to busy, and derive a deterministic video room name from the appointment id.

— TRY IT · RAG INGEST

Enumerate a patient's DocumentReferences, resolve content.attachment.url, fetch the Binary, then outline extract, chunk, embed, retrieve — with a DocumentReference citation on every answer.

— RECAP

● Quartet

● Flow

● Video

● RAG

— NEXT · CHAPTER 20

Everything we built creates obligations we have not paid: who joined the room, which documents entered the context, on what consent. Next: HIPAA, GDPR, the EHDS regulation, and the AuditEvent that records all of it.