

— CHAPTER 18 · AGENTS & MCP

Giving an LLM agent safe hands on FHIR.

A model can reason about a patient beautifully -- but only if it can reach the data. Clinical data lives in FHIR. This chapter is how an agent talks to it safely: typed tools, a consent gate, and an orchestration layer that knows when to pause for a human.

● MCP tools over FHIR

● Read vs write + human in the loop

● Mastra + BIRDY orchestration

— THE PROTOCOL · MODEL CONTEXT PROTOCOL

MCP: a host spawns one client per server.

● HOST

The agent runtime

Claude, or an agent runtime, spawns one client for each server it connects to.

● Data layer: JSON-RPC 2.0

● CLIENT

One per server

Each client holds one stateful connection. Messages are JSON-RPC two point zero.

● Tools / Resources / Prompts

● SERVER

Capabilities

Offers Tools, Resources and Prompts. Tools is the one this chapter uses.

● Current spec: 2025-11-25

● TRANSPORT

stdio or HTTP

stdio for a local process; Streamable HTTP for remote servers with auth.

— THE SHIFT · URLS VS TOOLS

The model picks a tool -- it never builds a URL.

— RAW HTTP · BRITTLE

The agent builds the request

The model constructs a URL -- GET Patient, name equals -- then paginates, parses a searchset Bundle, and untangles an OperationOutcome by hand. Every step is a chance to get it wrong, and it touches the token directly.

— TYPED MCP TOOL · CALM

The agent fills a schema

A named tool whose input schema says exactly which arguments are legal. The server holds the tenant and auth context, so the model never sees a token. Structured content comes back -- the model just reads it.

— ANATOMY · A TOOL WRAPPING A FHIR OP

A tool is four fields -- and a schema is a contract.

• NAME

search_patient

The handle the model invokes via tools/call.

• TITLE + DESCRIPTION

Human label

What the model reads to decide when to reach for it.

• INPUTSCHEMA

JSON Schema

Constrains the args -- search params, or a required resource id.

• DISCOVERY

tools/list

The host lists tools, then runs one with tools/call.

— ILLUSTRATIVE TOOLS WRAPPING HAPI

Tools like **search_patient** (read), **read_observation** (read), and **create_encounter** (write). Names are illustrative -- one named tool per clinical operation you want the agent to have.

— THE CONTRACT

The model cannot pass a parameter the schema rejects. The server takes the validated arguments, calls HAPI, and feeds the structured result back.

— THE SPLIT · FOUNDATION OF THE SAFETY MODEL

A wrong read wastes tokens. A wrong write is real.

— READ-ONLY · AUTO-RUNNABLE

search · read

No side effects. An agent can run them all day -- worst case it learns something it did not need. A hallucinated search costs a few tokens, nothing more.

— MUTATING · ALWAYS GATED

create · update · delete

Creating an Encounter, writing a MedicationRequest -- these change clinical state. A hallucinated medication order is a patient-safety incident. So the rule writes itself.

● Read-only runs unattended

● Mutating never runs without a gate

● Which is where the human comes in

— HUMAN IN THE LOOP · THE HOST OWNS THE GATE

The host must get explicit consent before any tool call.



Clinical agent

● Awaiting consent

Book a follow-up encounter for this patient next Tuesday.

The consent gate

MCP cannot enforce at the wire -- the host builds it



3

Elicitationspec-native confirm channel



4

Clinician approvesonly then does it write



5

Annotations untrustedunless the server is trusted

— CONCEPTUAL · A DUDOXX BRIDGE SERVER

The MCP server is just another well-behaved FHIR client.

ddx-fhir-r4-mcp

A local stdio process bridging the agent to HAPI



3

Bearer + X-Clinic-ID the same seam from Ch 13



4

HAPI :18080 the FHIR server we have built

— NOT A NEW AUTH SYSTEM

It reuses the bearer token and X-Clinic-ID seam from Ch 13, and the SMART scopes from Ch 17 still apply -- a read tool needs `patient/Observation.rs`, like any client.

— TENANT ISOLATION HOLDS

The partition interceptor from Ch 10 still isolates the tenant, even when an agent is the caller. The server gets no backstage pass -- the agent inherits the seams we already built.

— CONCEPTUAL · SHAPE, NOT A VERBATIM API

A workflow is the recipe; tools are the ingredients.

A clinical workflow

A durable sequence of steps over FHIR tools



3

Executor runs them, step by step

4

Reporter assembles the answer

— MASTRA · TYPESCRIPT AGENT FRAMEWORK

The workflow primitive composes steps in order, and each step can call one of our tool-backed FHIR operations.

— VERSION-SENSITIVE

The exact Mastra API moves quickly. Hold onto the shape, not the signatures. What matters: a workflow is a durable sequence, and any step can reach FHIR through a tool. Durable is about to matter a lot.

— CONCEPTUAL · THE MECHANISM BEHIND THE GATE

The read just runs -- the write waits, and nothing is lost.



Workflow runner

● Suspended

Summarize recent vitals, then create the follow-up encounter.

Suspend / resume

The human-in-the-loop gate, made real in the workflow



3 Surface decision clinician sees the proposed write



4 Resume on approval picks up exactly where it left



5 Write performed principle = consent, mechanism = this

— A DUDOXX PATTERN · NOT AN INDUSTRY STANDARD

BIRDY is the conductor; the MCP tools are the instruments.

• COORDINATOR

Receives intent

"Summarize recent vitals and book a follow-up" arrives as one clinical intent.

• DISPATCH

Routes to sub-agents

Each piece goes to a specialized tool backed by an MCP-exposed FHIR op.

• AGGREGATE

Collects results

It assembles what comes back into one coherent answer.

• GATE

HITL on mutation

On anything that writes, it applies the same human-in-the-loop gate.

— ORCHESTRATION ABOVE THE TOOLS

The tools, the read-write split, the consent gate, suspend and resume -- those are the parts. BIRDY decides which parts, in what order, for a given clinical intent.

— CONCEPTUAL · OURS, NOT A STANDARD

BIRDY is a Dudoxx-internal orchestrated-dispatch pattern. Frame it as the orchestration layer sitting above the ddx-fhir-r4-mcp tool surface.

— DEFENSE IN DEPTH · NO SHORTCUT FOR THE AGENT

The agent inherits the trust model, plus two human gates.

Five layers

An agent is the most powerful client -- so the most scrutiny



3 MCP consent consent + Elicitation gate the call



4 Suspend / resume holds every write for a human



5 AuditEvent + Provenance every mutation -- the whole of Ch 20

— NO NEW POWERS

The agent does not gain privileges. It inherits the same multi-layer trust model as any client -- plus two new human gates: consent, and the suspended write.

— FORWARD TO CH 20

Every mutation emits an AuditEvent and a Provenance record. That accountability layer is the whole of Ch 20 -- here it is the final safety net.

— SYNTHESIS · ONE REQUEST, START TO FINISH

Powerful where it is safe; paused where it counts.

The read path

Clinician prompt to Bundle



3 **stdio to ddx-fhir-r4-mcp**scopes + partition apply



4 **HAPI :18080**returns a Bundle

The write gate

Where the human sees the one moment that matters



3 **Clinician approves**the one moment they see



4 **Resume + write**the write lands



5 **AuditEvent emitted**scopes, partition, consent, audit

— TRY IT · THE SAFE HALF

Feel the tool-shaped response -- then wire it up.

— READ-ONLY SEARCH · PUBLIC SANDBOX

Point a search at the public HAPI sandbox:

```
hapi.fhir.org/baseR4/Patient?  
name=smith&_count=2
```

That structured result is exactly the shape an MCP read tool hands your agent.

— WIRE IT UP · CONCEPTUAL

The `claude.json` config takes a server stanza -- conceptually, an entry for a FHIR sandbox server:

```
~/.claude.json -> mcpServers: ddx-fhir-sbx
```

The shape you would configure, not a verified config.

● MCP tools over FHIR

● Read vs write + consent

● Suspend / resume + BIRDY

● Next: Ch 19 -- Telemedicine + calendar + RAG