

Building Agentic Dental Operations

For Fun and Profit

RISK MITIGATION SUMMARY · Sean Kubin | Standard Computer

The opportunity

Every practice carries unfinished business. Patients accept a treatment plan, life intervenes, and they never come back. Financial reporting sits in a CPA backlog. Team messages get overlooked.

That work is real revenue and real clinical value sitting in the practice's records. It is exactly the kind of work software agents are good at: patient, repetitive, detail-oriented, and never bored.

Agentic Dental Operations puts a small team of AI agents to work inside an environment built to handle patient information lawfully. Unfinished treatment recovery is the first use case. The same infrastructure supports whatever the practice wants to try next.

The shape of the solution

Three zones, with controlled interfaces between them. Nothing in Zone 1 is migrated, replicated, or replaced.

ZONE 1

The practice

The patient health information database and practice management system stay exactly where they are. They remain the system of record.

ZONE 2

The private cloud

A dedicated Microsoft Azure environment used only by this practice. It runs Paperclip — the agent platform — with org chart, work, budgets, credentials, and a permanent record of every action. No public address. Not reachable from the open Internet.

ZONE 3

The AI models

The reasoning capability the agents run on, accessed only under contracts that make the provider legally accountable for protecting patient information.

Zone 3 is contractually bound

Any vendor that handles patient information on the practice's behalf is a business associate under federal law. It cannot lawfully receive that information without a signed Business Associate Agreement. Both providers in scope offer one.

ANTHROPIC · CLAUDE

BAA covers the commercial API and Enterprise plans. It is executed by the organization's owner. HIPAA coverage is activated on the account, and zero data retention is enabled. Consumer subscriptions are not covered and are not used.

XAI · GROK

BAA is available through a request-and-review process. Enterprise terms require both a signed BAA and use of the zero-data-retention API before any patient information may be submitted. Coverage is not automatic and is not enabled by default.

Whichever models the practice chooses to run, **the agreement is signed before any patient data reaches them**, and the specific models and features in use are verified against what the agreement actually covers. Not every feature a provider offers falls inside its BAA.

The controlled interface

This is the most important piece of engineering in the design. It is worth being precise about what it is and what it is not.

It is not a copy of the database. No patient records are replicated into the cloud environment. The agents query the practice's systems through an interface and work with what comes back for the task at hand.

What is narrow is the interface, not the schema. The practice is expected to experiment — different systems, different fields, different workflows as the use cases evolve. The design does not assume a fixed set of fields it will never outgrow. What stays constant is the *shape* of the connection.

Every credential lives in a vault. API keys, tokens, and passwords for the practice management system, the messaging platform, the AI providers, and anything added later are held in an encrypted store inside the private cloud. They are injected only at the moment of use and never written into configuration files, scripts, tickets, or agent notes. The platform actively rejects attempts to store them any other way.

Credentials are scoped to specific agents and projects. Access is granted per agent and per project, not globally. The agent working recall campaigns holds the recall credential. It does not hold the billing credential, because it was never issued one. Adding a credential to an agent is a deliberate, approved act.

Every use is logged. The vault records which agent used which credential, when, and for which task. That log cannot be edited or deleted.

The connection surface is defined, not open-ended. Agents reach the practice's systems through a defined interface — an MCP connector or a purpose-built API endpoint — exposing named operations. They do not get a general database login, an admin console, or arbitrary query access. Widening the interface is a change that gets reviewed and recorded. The practice grows the system deliberately rather than discovering after the fact that an agent could reach further than anyone intended.

The principle

An instruction can be misread; an interface that was never built cannot be used. Written guidance shapes agent behavior. The vault and the interface constrain it. Both matter, and they are not interchangeable.

Direct patient contact

The system is designed to reach patients directly — scheduled recall messages, treatment follow-up, and campaign outreach sent without a staff member re-typing each one. That is the point of automating the work.

Direct contact raises the compliance stakes rather than lowering them, so the controls move into the pipeline itself.

Two categories of message, with different legal footing

A message about treatment a patient's own dentist proposed is a **treatment communication**. It is permitted without additional patient authorization, and it is where the highest-value work sits.

A message promoting products or services beyond that specific treatment — whitening packages, membership plans, financing offers, or anything a third party paid to include — is **marketing**. Marketing to identified patients using their health information requires prior written patient authorization. Automated calls and texts are separately governed by federal telemarketing law, which requires its own documented consent and carries per-message penalties.

The consent gate. Because a person is no longer reviewing each message before it leaves, contactability becomes something the system checks rather than something a human notices. Every outbound message passes a gate that verifies, per patient and per message type, that the required consent or authorization is on file and current, and that the patient carries no do-not-contact or deceased flag. No consent record, no send. The gate is enforced in the pipeline, not left to the sending agent's judgment.

Templates are approved, not improvised. Agents personalize within an approved template. They do not compose novel outbound copy from scratch. New templates and material revisions are reviewed against the treatment-versus-marketing standard and approved by the practice's Privacy Officer before going live. Every message sent is retained with its template version, consent basis, and timestamp.

Get this right the first time

Automating outbound contact without a consent gate is how a practice ends up sending thousands of unauthorized messages very efficiently.

How the environment is protected

Protection is layered. No single control carries the weight.

Three states of encryption

Most systems protect data in two states. This one protects three.

At rest. Everything stored is encrypted with keys the practice controls, held in dedicated security hardware. The storage platform cannot read it without those keys.

In transit. Every connection — practice to cloud, cloud to AI provider — travels inside an encrypted tunnel. Nothing moves in readable form.

In use. The uncommon one. The server runs on confidential computing hardware that keeps working memory encrypted while data is actively being processed, blocking the hosting platform and Microsoft's own administrators from reading patient information out of memory. On an ordinary cloud server, a sufficiently privileged platform engineer could in principle inspect memory. Here the hardware prevents it.

The perimeter

The environment has no public address. It cannot be found by scanning the Internet, because there is nothing there to find. Access requires a device individually approved and cryptographically enrolled into the practice's private network. Everything else is denied by default, and the server's own outbound connections are limited to an approved list — so even if something inside were compromised, it has nowhere to send anything.

Hardening

The server is built narrow rather than convenient.

Minimal surface. A stripped operating system with unnecessary services removed, patched automatically on a schedule that avoids practice hours.

Constrained software. Paperclip runs in a container with a limited access filesystem, stripped operating-system privileges, and hard limits on memory and processing. If compromised, it has very little room to move.

Pinned and scanned. Software is built from a fixed, recorded version and security-scanned before deployment. Nothing installs itself.

Isolated database. Working data lives in a managed database reachable only from inside the private network, encrypted under the practice's keys, with automated backups and tested restores.

The record

Every action is written to a log that cannot be edited or deleted: who accessed patient information, when, how many records, for which task, and what was sent to whom. Alerts fire on unusual access, failed logins, unexpected network activity, or any request returning more records than permitted. Logs are retained for two years in a separate store under an immutability policy.

If the practice is ever asked what happened, the answer exists.

How the agents are governed

The platform models a company. That structure is what turns intentions into enforceable controls.

The organization

Agents are arranged in a real hierarchy with reporting lines, not as a flat pool of chatbots.

A lead agent holds the practice's objectives and decomposes them into projects and tasks. It assigns work down the org chart and is accountable for the whole.

Specialist agents own domains — insurance verification, patient recall, campaign content, reporting, operations. Each has a defined role, a defined set of credentials, and a defined scope of work. They do not borrow each other's access.

A compliance reviewer sits outside the delivery chain and reviews any plan touching patient information or patient communication *before* work begins. It can block. The lead agent cannot override a block; a blocked item escalates to a person.

Delegation flows through the chart. When a task needs a capability an agent doesn't have, it routes to the agent that does, and the handoff is recorded. Work does not get done by an agent quietly acquiring new access.

Every agent has a budget. Hard monthly caps, enforced by the platform. An agent that reaches its limit stops. No surprise invoices, no runaway loops.

Approval gates on anything structural. Agents cannot hire other agents, expand their own permissions, change budgets, or add credentials without a person approving it. Autonomy is granted deliberately, one decision at a time.

The instruction layers

Agent behavior is shaped by written documents, layered from general to specific.

Company context — the practice’s mission, current objectives, and, where the practice has one, its policy manual. This is what lets an agent cite actual practice policy rather than improvising from generic knowledge.

SOUL.md — the identity layer. A per-agent document defining who the agent is: its role, its standards, its tone with patients, how it handles uncertainty, what it pushes back on, and its hard stops. This is what makes a recall agent sound like a member of the practice rather than a generic assistant, and it is the first thing loaded into the agent’s context on every run.

Role and project instructions — the procedural layer. What this agent does, in what order, with what data, and when it must stop and escalate.

Skills — reusable procedures for recurring work, so a task done well once is done the same way every time.

A note on what this layer is and is not. SOUL.md and its siblings are *persona and judgment* documents. They shape behavior reliably and are the right place to encode standards and tone. They are not security controls: they live in the model’s context window, they compete with other content for that space, and an unusual situation can pull an agent away from them. The design never relies on a written instruction where a vault, a scoped credential, an approval gate, or a consent check will do the job. **Instructions guide; architecture constrains.**

Inherited model safety

Underneath everything, the agents run on frontier models with their own safety training. Claude in particular is trained to a published set of behavioral principles and a usage policy, and declines categories of harmful requests independently of whatever instructions it has been given locally. In practice this behaves as a final backstop — an agent that has been misconfigured, or fed a malicious instruction hidden inside retrieved content, still carries a model-level disposition against acting on it.

This is a real benefit and worth understanding honestly: **it is a backstop, not a control the practice owns.** It is not configurable, not auditable by the practice, and not something to point at in a compliance review. It reduces the blast radius of a mistake. It does not replace the vault, the consent gate, the approval workflow, or the audit log. The compliance case rests on the layers the practice controls and can evidence.

What stays with the practice

Infrastructure is a foundation, not compliance. Compliance is roughly one-third technology and two-thirds documented policy and consistent behavior. These remain the practice’s responsibility:

- Designating a Privacy Officer and a Security Official in writing
- Maintaining a current security risk analysis covering the whole practice
- Maintaining the patient consent and authorization records the consent gate checks against
- Training staff and documenting it
- Approving message templates and any policy before adopting it
- Retaining compliance records for six years

IT supports all of it. None of it can be purchased “off the shelf”.

Delivery

Stage	What happens
1. Foundation	Azure environment provisioned, encryption keys established, network isolated, business associate agreements confirmed. Non PHI systems use hostinger.
2. Platform	Server hardened, Paperclip deployed, private access established between the practice and the cloud
3. Interface and vault	Credentials vaulted and scoped, connection to the practice's systems built with named operations, access logging verified
4. The organization	Org chart configured, SOUL and role instructions written, budgets and approval gates set, compliance reviewer wired into the workflow
5. Consent gate	Contactability and authorization checks built into the outbound pipeline, message templates approved and versioned
6. Validation	Full control checklist executed and recorded, restore tested, safeguards documented
7. First campaign	New Patient insurance benefits obtained and written to open dental. Unfinished treatment identified and outreach sent, probationary mode, under live monitoring.

Go-live rule

Nothing touches patient data until Stage 6 passes.

What the practice receives

- A dedicated, encrypted environment that only this practice uses
- A working agent organization recovering unfinished treatment, with room to take on more
- Business Associate Agreements on file for every vendor that handles patient information
- A **Safeguards Summary** listing every technical, physical, and administrative protection in place, with the regulatory requirement each one satisfies — written for the compliance binder, and the first document to produce in an audit
- Monthly evidence reporting and quarterly control review

This document summarizes the technical and operational design. A full implementation specification is maintained separately. Neither constitutes legal advice; the practice's privacy counsel should review the patient communication standards and consent requirements before go-live.

