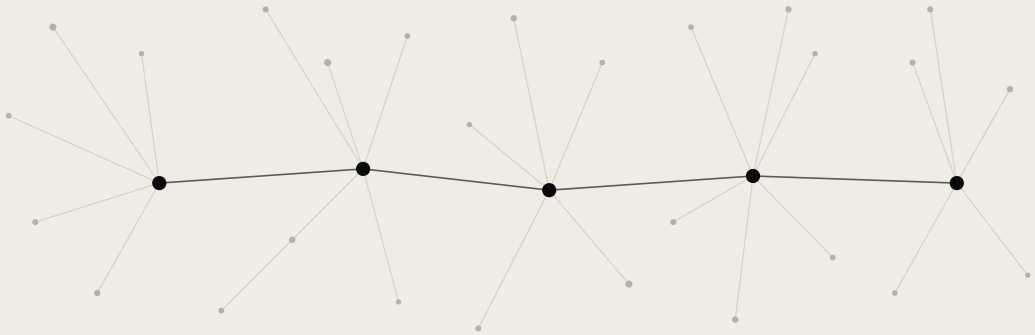
THE THESIS · AN ESSAY

Healthcare's Infrastructure Problem

Healthcare 2.0 — a manifesto for the infrastructure of continuous care.



TWENTY-NINE POSTS · FIVE MUSINGS · ONE ARGUMENT

NAING MAW · MELBOURNE · AUGUST 2026 · NAINGMAW.COM

1 · The wrong frame

Healthcare doesn't have a digital problem. It has an infrastructure problem.

The term "digital health" implies that healthcare is waiting to be digitised: that the solution is taking what exists and putting it on a screen. Better interfaces. Slicker apps. Faster forms. But the failure is not at the interface. It is in the plumbing underneath. The data layer that splits one patient into a dozen disconnected records. The prescribing loop that ends at the script. The care pathway that exists in a brochure and nowhere in the architecture. "Digital health" is a framing that accepts this foundation and tries to make it faster. The foundation is the problem.

The frame persists for a simple reason: interfaces are visible and infrastructure is not. An interface can be demonstrated in a meeting, screenshotted in a deck, funded in a round. A data model cannot. So the industry optimises for the layer that can be shown, and products get stacked onto broken workflows because rebuilding the workflow is slower, harder, and invisible right up until the moment it matters. The result is two decades of real improvements at the surface, with fragmentation dressed as innovation underneath.

The cost compounds quietly. Every health technology product that fails in its second or third year fails the same way: it was built on top of broken infrastructure rather than replacing it. The architecture was optimised for one use case, and the second use case always arrives. A second clinical context, a second care pathway, a second brand. The product bends, workarounds multiply, and the team discovers that the cost of supporting two contexts is nearly the cost of building a second system. Meanwhile the deeper cost lands on patients, and it is measured far too late. Health systems measure continuity by survey: did you have to repeat your history, did the follow-up happen. By the time a patient answers those questions, the decision that caused the discontinuity was made years earlier, in a data model that stores each episode in its own system and calls the gaps between them somebody else's problem. In that architecture the patient is the subject of the record, never its focus.

So the frame has to change. The patient is the unit of care, not the episode. Continuity is not a patient-experience metric; it is an architectural property, present or absent from the first schema onward. This is a building problem before it is a policy problem, a funding problem, or an adoption problem. And the people positioned to solve it are not the ones studying the system from above. They are the ones inside it.

I know this because I have spent thirteen years building against the same gap.

2 · What a decade of building taught me

The first thing mechanical engineering teaches you is that nothing fits perfectly, and that this is a decision, not a defect. Physical parts expand and contract. No machining is exact. So you design tolerances: deliberate leeway, engineered into every fit, so parts can move against each other without binding. A hinge swings not because every dimension is perfect, but because someone decided exactly how imperfect each one was allowed to be. That is what I actually took from an engineering degree in Singapore. Not the mathematics — the mindset. Accuracy is impossible. Flexibility is designed.

Clinical systems are built from the same truth, though almost nobody builds them that way. You cannot design for one hundred percent of cases, any more than you can machine a part to zero tolerance. A system designed only for the cases you imagined binds the moment reality expands past them. The discipline is deciding where the leeway lives: enough engineered flexibility to absorb the case nobody predicted, while keeping the whole tight enough to stay safe and efficient. And you learn where to look for failure: not at the obvious weak point, but at the joints, where one part hands its load to another. The prescribing system to the pharmacy. The record to the care team. The consult to the follow-up. You design for the worst case, not the average load, and you ask the only question that matters before touching anything: where does this fail? Not in the demo. In the field, under load, at scale.

The first build taught me what that question costs to answer. I joined a small Singapore company as an intern, servicing recycled smartphones and installing senior-friendly apps for underserved older adults. By the time I left in 2023, the company had been renamed ConnectedLife, I was running data and operations, and the system we had built was generating thousands of data points per household per day across a national deployment. We went from smartphones to wearables to full IoT kits: motion sensors in every room, a contact sensor on the front door, a hub collecting it all, deployed through Singapore's public housing programme as part of the Smart Nation initiative. The discovery that changed everything was how consistent a life is. An elderly person living alone follows remarkably stable routines, and a bathroom visit that usually takes fifteen minutes but stretches to forty is not noise. It is a clinically meaningful signal. Not complex AI; sustained, longitudinal observation of one life in one home. The clinical value was never in the sensor. It was in the pattern recognition across time.

Then scale arrived, and everything around the technology broke. The programme was designed for thousands of homes; the deployment reached roughly five hundred before the constraints

became clear. Installation broke first: every home needed a trained technician. Maintenance broke next: at five hundred homes, keeping the infrastructure running was itself an infrastructure problem. Uptake broke hardest: the person whose home was instrumented experienced the system as something that watched them, not something that helped them.

And underneath it all, the economics. That failure had no single moment; it was a realisation, arrived at by understanding the unit economics from inside them. We had priced the service at $25 to $30 a month with the hardware included, deliberately eating more than $700 of kit per home, on the bet that scale would absorb the cost. The bet wasn't wrong about scale. It was wrong about the year. In 2016, the only subscription most of the households we served had ever signed was a telco contract that came with a phone. We were asking them to adopt a pricing model that barely existed, for a service category that hadn't yet entered most people's minds. The architecture was sound. The clinical signal was real. The algorithm worked. The unit economics did not. The lesson took years to land properly: some infrastructure problems are not solved by moving faster. They are solved by staying long enough to watch the cost curves, the integration standards, and the clinical readiness converge.

The second build was smaller and cut deeper. In 2014, a team of us built a wearable for autism that nobody had asked for. We called it Heartware. The device sat at the chest and did something the clinical system had no frame for: it captured biometric data and environmental context simultaneously, at the moment of sensory dysregulation, and bridged the gap between why this is happening and what to do about it, in real time, in daily life. Published research backed the approach. A grant validated it. Three months later the co-founding team broke apart, before the prototype was finished. The IP stayed with me. The conviction went underground for twelve years. What makes the story worth telling is not the failure. It is what happened to the sketch. Persistent monitoring. Environmental context. Real-time feedback. Patient agency alongside caregiver visibility. The architecture we drew in a hackathon room is structurally identical to what the largest health platforms in the world are now racing to build. We were early. Not wrong.

The third build is the one running now: a multi-brand clinical platform in Australia, carrying prescribing, dispensing, patient records and multiple care pathways on shared infrastructure. The same gap, third time, in a different part of the pathway. A doctor writes a script; the pharmacy dispenses a substitute, or nothing; the record assumes the original; the next consult is grounded in a phantom data point. Same structural failure as a sensor gap in a Singapore flat: the space between clinical moments, where the data disappears. Except this time the curves have moved. The sensors are consumer-grade and accepted. The integration standards exist. The

economics that killed the first build are finally on the side of the people trying to solve the problem.

Engineering has a name for why these gaps are so hard to see: tolerance stack-up. The leeway that makes a single part work is exactly what accumulates, unwatched, across an assembly. Every component has a tolerance, and individually each is acceptable. But tolerances compound, and a system that passes inspection at every interface can still fail as a whole. Clinical platforms have exactly this failure mode. The booking system works, the prescribing system works, the dispensing integration works, and a patient falls through a gap that no individual system created but the combination made possible. The discipline that prevents it is the same whether the material is aluminium or software: design for the system, not the component. Resolve the complexity before it reaches the user. Test the assembly, not just the parts.

Singapore built the conviction. Toronto, at a distance, gave it language: a decade of building from inside a system became time to think about what the system should be. Melbourne is where it runs at scale. Three countries across thirteen years, and the settings changed while the thread did not. Healthcare infrastructure determines healthcare outcomes, and most people building in health tech are building the wrong layer. That pattern is only visible from inside the system. Which means the people who build the right layer will have to be the people who operate it.

3 · The operator advantage

There is a category of decision in healthcare infrastructure that only becomes visible from inside the system. Not through research. Not through consulting. Not through advisory boards or architecture reviews. Visible only to someone building the infrastructure and operating the clinical workflow at the same time. The default model in health tech keeps those two roles apart: one team builds from specifications, another team operates the clinic and works around what doesn't fit. The gap between them is where most healthcare technology quietly fails. Seven principles follow from collapsing that gap, and each was paid for in operation before it was written down.

Build from inside the system, not above it. A platform built for one clinic was copied wholesale to launch a second. Within months the prescribing workflows diverged and every feature had to

be built twice. The decision to consolidate into a single multi-brand architecture was not a technical insight available to an outside architect; it was an operational one, visible only to the person maintaining both codebases while running both clinics. The most consequential infrastructure decisions look like this: obvious in motion, invisible from above.

Build the load-bearing walls. Buy the furniture. Analytics dashboards, communication tools, file storage: furniture. Buy them, integrate them, move on. Prescribing, booking, patient records, dispensing integration: load-bearing walls. Every consult, every script, every patient interaction passes through them, and buying them means inheriting someone else's assumptions about how clinical decisions should work. Prescribing was the live version of this call. A capable external tool existed, buying it was the standard advice, and building looked like over-engineering. What settled the decision never appeared in an evaluation matrix: operating the clinics meant knowing how often prescribing rules actually change, and how tightly that logic couples to everything downstream of it. The wall got built, and every same-day rule change since has been the return on that call. The hardest part is knowing which is which, and that knowledge only comes from operating the system.

The system should change at the speed the clinic needs. Clinical rules are not static. A guideline changes, a product enters the formulary, prescribing rules need updating before the next morning's consults. With an external vendor, that change enters a queue: a ticket, a sprint, a release cycle that may or may not prioritise it. With an in-house system, it ships the same day. The difference is invisible on an architecture diagram and unmissable when you are the one fielding the call from the clinical team.

The automation boundary is a clinical decision disguised as a technical one. Draw it too far toward automation and the doctor becomes a rubber stamp. Too far toward manual review and the tool becomes noise. The boundary that works is: the system prepares, the clinician validates. That line could not be drawn by an engineer optimising for capability or a clinician optimising for safety; it required someone holding both. Chapter 4 returns to this boundary, because it turns out to be the keystone of the whole architecture.

Design for the clinician who has no time, not the one in the demo. "I don't have time. Just tell me when something is wrong." That sentence, from a working clinician, is not a feature request. It is an architectural constraint, and most clinical software is built in violation of it.

The operator advantage compounds. The consulting advantage expires. Consulting produces a point-in-time assessment: accurate the day it is delivered, decaying from then on. Operating

produces a continuously updated model of how the system works, where it breaks, and what needs to change next. The first is useful. The second is irreplaceable, and it is the asset that grows simply by staying inside.

Healthcare infrastructure must be built from the inside out. Not because inside knowledge is superior in principle, but because the decisions that determine whether a clinical system actually works are only visible from inside it.

The honest counter-argument deserves its place: operators risk tunnel vision. In-house bias, over-engineering, building what could be bought, mistaking familiarity for insight. The discipline that answers it is already in the list. Build the load-bearing walls, buy the furniture: the operator advantage is precisely the ability to tell the difference, and that ability is itself the check on over-building. An operator who builds everything has stopped operating and started collecting; the principle cuts both ways, and it is meant to. The deeper check is in the story itself: the same problem approached from more than one position. The final chapter returns to it.

The operators who build, and the builders who operate, are the ones who will build healthcare infrastructure that holds. What follows is what that infrastructure actually looks like.

4 · The architecture that holds

Continuity of care is a data-architecture problem before it is anything else. Everything in this chapter is one expression of that sentence: six engineering choices that look unrelated and are in fact the same constraint, applied six times.

One owner per truth. In a sound clinical estate, every fact has exactly one authoritative home. The operational platform owns the clinical record. The relationship system owns the relationship. The finance system owns the money. The analytical layer owns nothing: it holds copies, shaped for questions, and when its numbers disagree with the operational system, the operational system is right and the pipe gets fixed. This sounds like bookkeeping. It is continuity. The moment a reporting tool quietly becomes a second system of record, two versions of the patient exist, and history can no longer carry forward without contradiction.

The longitudinal model. The patient is the entity, not the episode. Most clinical systems store a collection of episode-specific records, each starting from near zero, and then bolt on integration

projects to reassemble what the original architecture broke. The alternative is a data model where history carries forward by default because it was never split in the first place: one record, accumulating across consults, prescriptions, results and daily life, with the episode as a view over it rather than the unit of storage.

Translation at the edge. The interoperability standard is a wire format, not a storage format. The wrong way to adopt it is to rebuild the internal data model in its image, which puts a messaging standard in charge of your database design and trades a sharp internal model for the lowest common denominator of world healthcare. The right way is to keep the internal model exactly as sharp as the clinic needs and translate at the boundary, in one adapter, so that when the standard churns, and it churns, the blast radius is the adapter, not the platform. Clinical truth stays home; interoperability is how it travels.

Silent failures are patient-safety failures. Most software fails visibly: a crash, an error, a complaint. Clinical systems fail silently, and the dangerous failures look like normal operations. The monitoring platform that assumes a clinician is reviewing the data when nobody has time to look. The symptom log treated as ground truth when patients estimate and forget. The wearable sync that drops silently, leaving a gap in the record that reads as inactivity when it is actually a broken connection: a patient flagged as fine because the system reported green. The fix is not another dashboard. Clinicians have said precisely what they need: actionable alerts, surfaced inside the system they already work in. That is an architectural choice. The platform owns the alerting layer and is designed around the reality that nobody has time to look, not the assumption that someone will. And when the fix works, it works silently too. Weeks after dispensing data began flowing directly into the patient record on a live clinical platform, the clinical team had nothing to say about it. Doctors opened the consult, saw what the pharmacy actually dispensed, and moved on. No feedback, no tickets. Months of integration work, to produce a moment the user doesn't register. The highest praise a clinical system can earn is silence.

The automation boundary, resolved. The design constraint in clinical AI is not accuracy. It is the boundary between preparation and decision. A system that prepares information for a clinician to review is infrastructure; a system that delivers clinical guidance directly to a patient is practising medicine without a licence, however good the model. The resolution has two halves. The clinician is always in the loop: permanently, not as a transitional safeguard to be removed when the model improves. And the determination of "safe enough" is deterministic, not generative: a hard rail that runs independently of the model and never asks the model to judge its own safety. Together they form a constitution, a small set of inviolable laws sitting above the

intelligence. You write the constitution once; the model gets smarter underneath it forever. What changes as the system improves is not whether a human decides, but how well the system protects the human's attention for the decisions that genuinely need one.

Built slowly, on purpose. Speed in healthcare infrastructure is not how fast you ship. It is how long what you ship holds. The fast company launches first, gets traction, and bends when the second use case arrives; workarounds multiply until the rebuild. The slow company spends its early time on the data model, designs the prescribing layer for the second brand before the second brand exists, and looks from the outside like it is behind. Then the second use case arrives and the architecture absorbs it, the third arrives and the platform holds, and the speed advantage inverts: every change is a configuration, not a rebuild.

Six choices, one constraint. Each is continuity of care, expressed as an engineering decision rather than a slogan. Build this layer correctly, and something larger than a better clinical platform becomes possible: a different kind of clinic altogether.

5 · What comes next

For ten years I believed continuous care was waiting on three curves. Sensor cost and acceptance: crossed. Wearables are cheap, accurate, and worn by choice; the question that killed a $700 hardware kit in 2016 is settled. Clinical validation: moving, visibly. Reluctance, not refusal, and the trajectory is set. The third curve was the one I misread. Continuous data was never hard to collect. It was hard to watch. A stream of daily signals demands attention nobody in a clinic has; that is why every monitoring dashboard of the last decade was ignored. That curve moved in the last two years. AI can now carry the first level of analysis: reading the stream, learning the baseline, surfacing the anomaly. Not making the clinical call. Preparing it. The same boundary that constrains AI in Chapter 4 is the thing that unlocks continuous care here. The convergence is not three curves crossing at once. It is the last and hardest one finally moving.

When that layer exists, the interface starts to disappear. The measure of maturity is not adoption; it is disappearance. Most days the system says nothing, and the silence is the product working. A duck on water reads as still from the shore while it paddles constantly underneath; the right healthcare system runs the same way. When it speaks, it says one true, clinician-validated thing, drawn from the person's own record at the moment it matters, then gets out of

the way. Engagement stops being a success metric and becomes a cost: every surfacing is a claim on someone's attention, and the system should be judged on what each claim was worth. The phone stops being a dashboard demanding interpretation and becomes a digital extension of the patient.

The structural consequence is the one almost nobody building in this space will say plainly: the consult becomes the exception. Today the appointment is the toll paid for any change, and everything between visits is invisible. In the continuous model the background is the default. Safe things proceed and are logged; contested things rise to a clinician: continue, adjust, or bring them in. The visit is reserved for what genuinely needs one, and it arrives prepared. This is a business-model shift hiding inside a product decision. Clinic economics, and the funding models around them, assume the consult is scarce and mandatory; loosening that changes the unit economics of care, and it should be planned with clinical and financial leadership in the room rather than discovered after the fact. It will be the most contested claim in this essay, and it is where the next decade of clinical-model innovation lives.

And underneath it all sits the record: longitudinal, portable, built to outlive any single clinical relationship. The durable thing a healthcare company can build is not the relationship, which can and should be free to end. It is being someone's base layer for life: the ground their care stands on wherever they go. Portability stops being a retention threat and becomes the point. Build it so she could leave, and she chooses to stay.

The obvious objection is already standing in the middle of the space: the player who owns the wrist and the data, the best-resourced and best-positioned company in health technology, has held a base layer for a decade and not won healthcare with it. But look at which layer it owns. The wrist and the data are the consumer base layer. The base layer that matters in healthcare is the one nearest the clinical decision, and reaching it means building and operating the clinical system itself: the one thing they will not cross into. That is not failure; hardware and data remain enormously profitable layers. It is a ceiling, chosen. In a decade where advantage compounds for whoever stands closest to the clinical decision, the asset does not win healthcare if you will not move to where it becomes care.

I know where that boundary sits because I have approached it from every side. First the device: Heartware, the bet that hardware could be care. Then software sold into the system: the Singapore years, selling continuous care in from the outside. Now the system itself: building and operating the clinical platform, every prior lesson inside it. Three positions, thirteen years, one problem. It is also the honest answer to the operator's failure mode: the protection against

inheriting a system's blind spots is not an outside consultant; it is having stood outside the system, sold into it, and finally built it, remembering what each position showed. Continuity of conviction, not continuity of vantage point.

Which brings me back to a device that sat at the chest, close to the heart. Heartware was never really a wearable. It was a bet that the missing layer in healthcare was clinical integration: the architecture connecting what a device measures to what a care team does. That layer didn't exist in 2014, so the device couldn't matter. Build the layer, and the device stops being a gadget and becomes care. Early, not wrong. And the infrastructure to prove it finally exists.

It is worth saying plainly what all of this was for. The tools of medicine have improved beyond recognition; the model of care has not. Reactive, episodic, population-level — that model is centuries old, and inside it the patient has always been the subject of healthcare, never its focus. That is what changes now. When the base layer is right, the unit of care shifts from the population to the person, and the work from treating disease to preventing it.

Somewhere, a woman makes tea while her phone lies face-down on the bench. It lights once, with one quiet line from a system that has known her for years and a clinician who trusts what it prepares. She glances, half-smiles, and gets on with her morning. No appointment was needed. Nothing was performed. Care simply happened, the way infrastructure works when it is finished: invisibly. Still from the shore, and never once not moving.

That conviction has not changed in thirteen years. It will not change in the next ten.

NAING MAW · AUGUST 2026
THE LIVING VERSION OF THIS ESSAY: NAINGMAW.COM/WRITING/HEALTHCARES-INFRASTRUCTURE-PROBLEM