

# How Tampa General & Palantir Built a Sepsis Detection System That Saved 886 Lives

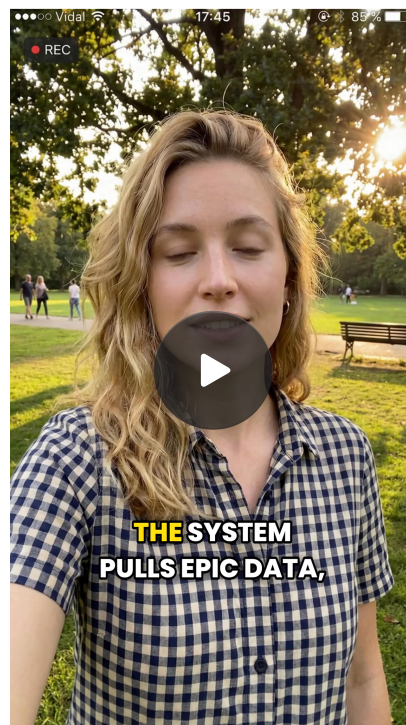
How a Solo, Bootstrapped, Vibe-Coding Entrepreneur Could Build the Same Thing Inside Any Hospital Without the Nine Figure Contract

JUN 14, 2026



Share

## Video Preview



## Part I Podcast free on Spotify.



Thoughts on Healthcare Markets & Technology



### **Part I: How Tampa General & Palantir Built a Sepsis Detection System That Saved 886 Lives**

Tampa General cut early sepsis deaths 68% and saved 886 lives using Palantir. The algorithm is not the reason why...

▶ Listen now

14 days ago · Thoughts on Healthcare

## **Part II Podcast episode for paid subscribers only. Also on Spotify.**



Thoughts on Healthcare Markets & Technology



### **Part II: How Tampa General's Sepsis Detection System That Saved Lives**

Tampa General cut early sepsis deaths using Palantir. The algorithm is not too far from what we built.

▶ Listen now

14 days ago · Thoughts on Healthcare



### **Discover more from Thoughts on Healthcare Markets & Technology**

Expert analysis of healthcare and life sciences markets, technology, investment, entrepreneurship policy, and AI — for investors, entrepreneurs,...

Over 4,000 subscribers

Enter your email...

**Subscribe**

By subscribing, you agree Substack's [Terms of Use](#), and acknowledge its [Information Collection Notice](#) and [Privacy Policy](#).

Already have an account? [Sign in](#)

*To listen to paid episodes in Apple or Spotify, link to those settings on those platforms (instructions inside the Podcast).*

## **Table of Contents**

1. Abstract
2. What Tampa General actually built, best guess
3. The FDE model, or why Palantir wins deals it has no business winning
4. The solo FDE pitch: scope, contract structure, rate
5. Procurement: what to buy and what to beg for
6. Data plumbing: feeds, normalization, mapping
7. The algorithm is the easy part
8. Workflow is where sepsis tools go to die
9. Validation and governance, or how to not kill anyone
10. Training and rollout
11. Will Epic compete
12. The economics, and why this is a real business

# Abstract

- Tampa General Hospital plus Palantir Foundry: sepsis surveillance hub credited with 886 lives saved since Aug 2022, 68% drop in early sepsis mortality, ~30% shorter LOS for sepsis patients, ~1,000 beds monitored in real time
- Core thesis: the value was never the algorithm. It was data unification plus workflow plus a human rapid response loop. All three are replicable by one skilled contractor embedded inside a hospital
- This piece reverse engineers the build: ADT/ORU/lab feeds, terminology mapping (LOINC, RxNorm, SNOMED), a scoring layer that is closer to logistic regression than LLM magic, alert routing to a dedicated team, and the SEP-1 bundle as the downstream payoff
- Covers procurement (almost nothing), interface engine realities, Epic Sepsis Model's documented failure (AUC 0.63 external validation), Epic's competitive response, and unit economics for a solo FDE charging hospital rates
- Audience: operators, investors, and clinicians who already know what an ADT feed is and want the build sheet

## What Tampa General actually built, best guess

Start with the headline numbers because they are genuinely wild. The system is estimated to have helped save 886 lives since August 2022, and internal analysis shows early deaths from sepsis down 68% with sepsis patients spending about 30% less time in the hospital. The platform pulls electronic health record data, lab results, clinician notes, and bedside monitor output into a centralized dashboard tracking roughly 1,000 patients at once, and when patterns suggest early sepsis it alerts a rapid response team. Flagged patients get antibiotics within an hour. The Sepsis Hub is one of more than 60 tools the hospital has built on Palantir's software since the collaboration started in 2021.

Now strip the press release varnish and ask what is actually running under the hood. Foundry is not a sepsis product. Foundry is an ontology layer, a pipeline orchestration tool, and a dashboard builder that Palantir engineers configure on site. So the honest description of the Tampa General build is: a data integration project that took Epic Chronicles/Clarity data, HL7v2 feeds, and device telemetry, normalized it into a patient object model, ran a continuously evaluated risk score against it, and wired the output into a paging and huddle workflow staffed by actual humans. The 68% mortality reduction is not a model performance stat. It is a process stat. Antibiotics

within an hour of suspicion is the entire ballgame in sepsis, where every hour of delayed abx in septic shock costs somewhere around 4 to 8% mortality depending on which Kumar era study you trust. The model just decides who gets looked at. The rapid response team and the abx clock do the saving.

That distinction matters enormously for anyone thinking about replicating this, because it means the moat is not a proprietary algorithm trained on some secret dataset. The moat is an engineer sitting in the hospital for 18 months making the pipes work and making nurses not hate the alerts. Which is exactly the thing a solo operator can sell.

One more piece of context. TGH is a roughly 1,000 bed academic-affiliated system, an Epic shop, with a chief data and analytics officer who explicitly describes the architecture as sitting Palantir's toolsets on top of the data coming from the electronic patient record and from machines. That phrase, on top of, is doing a lot of work. Nothing got ripped out. Epic stayed the system of record. Palantir became the system of computation and the system of attention. Hold that thought for the Epic section.

## **The FDE model, or why Palantir wins deals it has no business winning**

Palantir's actual product, the thing that justifies the contracts, is the forward deployed engineer. The FDE shows up, gets badged, gets VPN access, sits with the clinical informatics team, and builds the thing in situ instead of shipping software and a PDF. Every hospital that has ever bought enterprise clinical software knows the failure mode: vendor delivers, integration drags for 14 months, the thing goes live half configured, clinicians ignore it, the license renews anyway. The FDE model fixes this by collapsing the vendor and the integrator into one person who is accountable for the outcome, not the deliverable.

The dirty secret is that the FDE does not need Foundry. Foundry gives the FDE leverage, sure, nice pipeline tooling, a decent ontology abstraction, pretty dashboards. But the FDE's real toolkit is Postgres, Python, an interface engine, and the social capital to get the CMIO to answer Slack messages. Palantir charges seven to eight figures annually for health system deployments. The labor component of a single use case build, the sepsis hub specifically, is plausibly one to three engineers for a year. The arbitrage is sitting right there in the open.

This is the whole pitch for the solo FDE: be the Palantir engineer without the Palantir invoice. Hospitals in the 200 to 600 bed range, which is most of American inpatient capacity, will never be a Palantir customer. The deal size does not pencil for Palantir's

sales motion. But those hospitals have the same sepsis mortality problem, the same SEP-1 compliance pressure from CMS, the same fragmented data, and CFOs who can absolutely find \$400K for a project with a mortality and LOS story attached. That is the white space.

## **The solo FDE pitch: scope, contract structure, rate**

The buyer is usually the CMO or CQO, with the CMIO as champion and the CISO as the boss fight. The pitch is not AI sepsis detection. The pitch is: your sepsis mortality index is X, your SEP-1 bundle compliance is Y%, your sepsis patients are costing you Z excess days, and there is a documented playbook from a 1,000 bed Florida system that cut early sepsis deaths by two thirds. One embedded engineer, 12 months, fixed milestones, you own all the code and IP at the end. No SaaS lock-in, no per-bed licensing, no data leaving your tenant. For a quality executive who has been burned by vendors, the you-own-everything line is the closer.

Contract structure should be an independent contractor agreement under the hospital's standard professional services template, with a BAA executed because the contractor is unambiguously a business associate touching PHI. Workers comp style co-employment risk is low if the contractor brings their own equipment, sets their own hours, and bills against milestones rather than punching a clock, but the hospital's counsel will have opinions and the path of least resistance is sometimes routing through an existing staffing MSA or forming a single member LLC with its own liability policy. Carry cyber liability and E&O, minimum \$2M, because a contractor with hands on a clinical pipeline without insurance is a contractor who does not understand the room.

Rate: clinical data engineering with informatics fluency bills \$175 to \$300/hr through staffing firms, and the firm keeps a third. Direct, a credible solo FDE quotes \$200 to \$250/hr or a fixed annual engagement in the \$350K to \$500K range with milestone payments tied to feed validation, silent mode go-live, pilot unit go-live, and house-wide rollout. Anchor against the alternative: Palantir at \$5M+ a year, or an Epic optimization consultancy that will charge \$300K to configure a tool that, as covered below, demonstrably underperforms.

## **Procurement: what to buy and what to beg for**

Here is the part that surprises people: the shopping list is nearly empty. Almost everything required already exists inside the hospital, gathering dust or locked behind a committee.

Compute: a few VMs in the hospital's existing VMware cluster or its Azure/AWS tenant. A sepsis surveillance workload across 500 beds is, computationally, a joke. Streaming a few hundred HL7 messages per minute, maintaining state for active encounters, scoring on a cadence of every 15 minutes or on new-data triggers. A 16 core box with 64GB of RAM is overkill. Budget maybe \$2K/month of cloud spend if going managed, or zero marginal cost on-prem.

Software: Postgres or SQL Server (already licensed), Python, an open source interface engine like Mirth Connect (now Open Integration Engine post the NextGen licensing drama, and the licensing change is worth checking current state on before committing), or the hospital's existing Rhapsody/Cloverleaf/Epic Bridges capacity. Dashboarding via whatever the BI team already runs, Power BI or Tableau, or a lightweight internal web app if the team can support it. Grafana works shockingly well for ward-level vitals walls and nobody talks about this.

What needs to be begged for, and this is the actual procurement battle: an outbound HL7 feed allocation from the interface team (interface engineers are the scarcest resource in any hospital, their backlog is measured in quarters, and getting two weeks of their time requires executive sponsorship), read access to Clarity or Caboodle if it is an Epic shop, a service account with appropriate scoping, network segmentation approval from security, and, critically, a slot on the clinical decision support committee agenda. The committee slot takes longer than the engineering. Plan for 60 to 90 days of governance latency before the first byte of PHI flows, and use that time to write the data dictionary and the validation protocol so the committee has something to approve.

## **Data plumbing: feeds, normalization, mapping**

The data architecture for a sepsis surveillance system needs five streams, and anyone who has built a hospital data product can recite them in their sleep. ADT messages for census, location, and demographics, because knowing who is in which bed right now is somehow still the hardest problem in healthcare. ORU messages for lab results: lactate, WBC, creatinine, bilirubin, platelets, blood cultures ordered and resulted. Vitals, which arrive either as ORU feeds from the device integration layer (Capsule, now Philips, or Cerner CareAware, or Epic's device integration) or get scraped from flowsheet rows in the EHR, and the flowsheet path introduces charting

latency that matters clinically since a nurse charting vitals 45 minutes after taking them eats your entire detection lead time. Medication administration records, because knowing whether abx have already started changes the alert logic entirely. And orders, to detect that a clinician already suspects sepsis (cultures plus lactate ordered together is itself a signal).

Real-time HL7v2 over the interface engine is the workhorse. FHIR is the polite modern answer and the subscription/bulk APIs keep improving, but in 2026 the v2 feeds remain more complete, lower latency, and better understood by the interface team, so the pragmatic build is v2 for streaming plus FHIR or Clarity for backfill and enrichment. Pull two to three years of historical encounters from Clarity for model calibration and baseline mortality/LOS measurement. That retrospective extract is also the political deliverable: showing the CQO their own sepsis cohort, properly defined, with bundle compliance and mortality stratified by unit, buys six months of goodwill before anything goes live.

Normalization is where the FDE earns the rate. Every hospital's lab catalog is a crime scene. Lactate exists as four different local codes across the main lab, the POC analyzers, and the ED iSTATs, with different units (mmol/L vs mg/dL, and yes both will appear). Map everything to LOINC, meds to RxNorm with an antibiotic class table, problem lists and diagnoses to SNOMED/ICD-10, and maintain a local-to-standard crosswalk table as a first class artifact with an owner, because the lab will add a new analyzer in month seven and silently break a feed, and the only defense is automated feed monitoring that alarms on volume anomalies per code. Unit conversion bugs in clinical pipelines are the kind of thing that ends careers, so every numeric gets a plausibility range check at ingestion and out-of-range values get quarantined, not coerced.

Then build the patient state object: one row per active encounter, continuously updated, holding latest and trending values for every input, time since last culture, abx status, comfort care flags (do not page the rapid response team about a hospice patient, this single filter is worth more to clinician trust than ten points of AUC), and unit context, because a lactate of 3 means something different in the ICU than on a med-surg floor.

## **The algorithm is the easy part**

Everyone wants to talk about the model. The model is genuinely the least interesting component, and the literature says so loudly. The clinically validated options, roughly in ascending order of sophistication: SIRS criteria (sensitive, hilariously nonspecific, fires on every post-op patient with a fever), qSOFA (specific, too late, by the time

qSOFA trips the horse has left and burned down the barn), NEWS2 (the NHS workhorse, decent general deterioration signal), and then the learned models. Hopkins published TREWS results in Nature Medicine in 2022 showing that alerts confirmed by providers within three hours were associated with meaningful mortality reduction, and the underlying machinery was gradient boosting on a few dozen features. Duke's Sepsis Watch ran deep learning but the published lesson from that program was overwhelmingly about the workflow and the rapid response nurses, not the architecture.

A defensible v1 for a community hospital: gradient boosted trees (XGBoost, LightGBM, whatever) on 40 to 80 features covering vitals levels and deltas, lab values and trajectories (delta lactate matters more than absolute lactate), demographics, comorbidity flags, and care context features like recent surgery or central line presence. Train on the hospital's own retrospective cohort with sepsis labels derived from the CDC Adult Sepsis Event definition rather than billing codes, because ICD-10 sepsis coding reflects reimbursement incentives at least as much as clinical reality and everyone in this audience knows exactly why. Calibrate the threshold not for AUC but for alert burden: the design constraint is alerts per nurse per shift, and the honest number a floor can tolerate is low single digits. A model with AUC 0.85 firing 30 times a shift loses to a model with AUC 0.78 firing four times a shift, every time, forever.

NLP on notes, which the Tampa General write-ups mention, is real but secondary: nursing notes contain early soft signals (confused, lethargic, looks unwell) hours before vitals move, and a lightweight classifier over note text adds lead time. Fine addition for v2. Anyone proposing an LLM as the primary detector in v1 should be escorted from the building, partly for latency and cost reasons, mostly because the validation and governance burden for a non-deterministic scorer in a life-critical loop is not a solo contractor problem to want.

Run the scorer on every new relevant data event plus a 15 minute heartbeat. Persist every score with full feature snapshot for auditability. That audit trail is not optional, it is the thing the quality committee, the malpractice carrier, and eventually maybe FDA-adjacent scrutiny will ask about.

## **Workflow is where sepsis tools go to die**

The graveyard of sepsis alerting is enormous and well documented, and every corpse died the same way: alert fires to the bedside nurse or into the chart as a BPA, nurse is managing six patients, alert joins the other 40 interruptions that shift, click, dismiss,

repeat until the override rate hits 95% and the tool is dead tissue. Epic's own BPA-based sepsis alerts historically suffered override rates in exactly that range.

Tampa General's design choice, routing alerts to a dedicated rapid response function rather than the bedside, is the load bearing decision in the whole system. A small team, often a sepsis coordinator or RRT nurse, owns the alert queue. They triage on the dashboard, do a two minute chart review, call the floor or walk to the bed, and activate the huddle if it is real. This converts a noisy classifier into a precise human-mediated escalation, and it means the bedside nurse only ever gets contacted by a colleague with context, never by a popup. The economics: one dedicated FTE per shift covering a few hundred beds, maybe \$300K to \$450K a year fully loaded for round the clock coverage shared across roles. Against the avoided cost of even a handful of ICU upgrades and the LOS reduction, the staffing math clears trivially, but it has to be in the proposal because a hospital that buys the software without funding the team has bought a screensaver.

Downstream of the huddle is the SEP-1 bundle, the CMS quality measure: lactate, blood cultures before abx, broad spectrum abx, fluids for hypotension or lactate over 4, repeat lactate, all on defined clocks. The dashboard should track bundle element completion per flagged patient in real time, with timers, because SEP-1 moved into value-based purchasing and bundle compliance now carries direct payment consequences. This gives the CFO a revenue-adjacent metric while the CMO gets the mortality metric, and a project that feeds two executives' scorecards simultaneously is a project that survives budget season.

## **Validation and governance, or how to not kill anyone**

Silent mode is non-negotiable. Run the model live for 8 to 12 weeks generating scores that no clinician sees, then adjudicate: pull every patient who would have alerted, have a clinician panel review timing against actual sepsis onset and treatment, measure lead time versus first clinician action, count the would-be false alarms per unit per shift. This produces the sensitivity/PPV/lead-time package the CDS committee needs and surfaces the embarrassing failure modes (dialysis patients, liver patients with chronically cooked labs, L&D) before they surface themselves at 3am.

Regulatory posture: a locked, transparent, institution-deployed predictive model with clinician-in-the-loop currently lives in the zone of clinical decision support that FDA has mostly not enforced against under the 21st Century Cures CDS carve-out, particularly when the basis for recommendations is explainable and the clinician can independently review. The contractor should not market it as a device, should

document the carve-out analysis, and should put model governance, performance monitoring, drift detection, quarterly recalibration review, a named clinical owner, in writing. ONC's HTI-1 transparency requirements for predictive decision support in certified health IT also shape what attributes need documenting. None of this is hard. All of it is the difference between a professional and a tourist.

## **Training and rollout**

Rollout sequencing that works: one pilot unit, usually a med-surg floor with a motivated nurse manager, never the ED first (the ED has its own screening culture and its own politics). Four weeks on the pilot floor with the FDE physically present for day shift, tuning thresholds against real feedback, then stepwise expansion floor by floor over a quarter. Training is not an LMS module. It is the sepsis coordinator and the FDE doing 15 minute huddles at shift change, showing the dashboard, explaining what the score does and pointedly what it does not do, and committing publicly to a feedback loop where every alert a nurse flags as dumb gets reviewed and answered within a week. Clinician trust is purchased one acknowledged false positive at a time. The Hopkins TREWS papers are explicit that adoption and confirmation rates, not model metrics, drove the mortality result, and the adoption was earned through exactly this kind of grind.

Total timeline, realistically: month one to three governance and access, month three to five pipelines and retrospective build, month five to seven silent mode, month seven to eight pilot floor, month nine to twelve house-wide. Twelve months, one FDE, with the interface team and a clinical champion as part-time force multipliers.

## **Will Epic compete**

Yes, and it already lost once in public, which is the fun part. The original Epic Sepsis Model, deployed at hundreds of hospitals as a packaged predictive tool, got externally validated by Michigan in JAMA Internal Medicine in 2021 and the results were brutal: AUC around 0.63, missed two thirds of sepsis cases at the deployed thresholds, generated huge alert burden, and the affair became the canonical case study in why proprietary opaque clinical models need external validation. Epic subsequently rebuilt the model, and to be fair the newer version trained differently and the company has been shipping serious platform capability since: Cognitive Computing, the Cosmos dataset, generative features, and Epic will happily tell any CIO that sepsis prediction is included in what they already pay for.

So the competitive answer is nuanced. Epic will absolutely ship a sepsis score. What Epic does not ship is the rest of the system: the device-latency-aware vitals pipeline,

the dedicated response team workflow, the local calibration, the silent mode validation against the hospital's own cohort, the SEP-1 timer dashboard wired to the huddle, the human standing on the unit at shift change. Epic sells configuration surface. The outcome requires deployment labor that Epic structurally does not provide and that Epic's consulting ecosystem provides expensively and generically. Tampa General is the proof: an Epic shop, with access to everything Epic sells, that chose to put a third party computation layer on top and staff it, because the packaged option was not producing the outcome. The solo FDE is not competing with Epic's model. The solo FDE is competing with the hospital doing nothing, which historically wins about 80% of these deals, and that is the actual enemy.

The realistic long game: Epic keeps improving, packaged scores get good enough that the model layer fully commoditizes, and the durable business is the deployment and workflow layer, which conveniently is the part the FDE owns. There are worse positions than selling the labor that makes the incumbent's free feature actually work.

## **The economics, and why this is a real business**

Close with the napkin. A 400 bed community hospital sees maybe 1,500 to 2,500 sepsis cases a year depending on case mix and how honestly they are coded. Sepsis carries roughly 10 to 20% inpatient mortality and excess LOS measured in days, with per-case costs that make it the most expensive condition in US hospitals at somewhere north of \$38B annually nationwide and rising. If a surveillance program at that hospital replicates even a quarter of Tampa General's effect, the avoided mortality is dozens of lives a year and the LOS savings alone, at \$2K to \$3K per avoided bed day across a few thousand cases shaving fractions of a day on average, runs into seven figures. Against a \$400K contractor engagement plus \$400K of response team staffing plus rounding-error infrastructure, the payback period is measured in months, and the mortality story writes the board presentation by itself.

For the contractor, the model scales the way good services businesses scale: first deployment is the reference, second and third reuse 70% of the pipeline code and 100% of the playbook, and by deployment four the engagement compresses from twelve months to seven, at which point the choice is run two concurrent and gross \$700K to \$900K solo, or hire the second FDE and become the thing Palantir was before the stock ticker, a deployment shop with a product slowly crystallizing out of repeated builds. The crystallized product, a portable sepsis hub that installs against any Epic or Oracle Health shop in a quarter, is a venture-scale company. Several have tried. The ones that failed mostly sold software. The lesson from Tampa, from

Hopkins, from Duke, is that the winners sell the outcome, and the outcome lives in the building.

---

## Subscribe to Thoughts on Healthcare Markets & Technology

By Thoughts On Healthcare Markets & Technology · Hundreds of paid subscribers

Expert analysis of healthcare and life sciences markets, technology, investment, entrepreneurship, policy, and AI — for investors, entrepreneurs, hospital and insurance executives, and physicians navigating the business of healthcare.

By subscribing, you agree Substack's [Terms of Use](#), and acknowledge its [Information Collection Notice](#) and [Privacy Policy](#).



2 Likes

[← Previous](#)

[Next →](#)

## Discussion about this post

**Comments**

Restacks



Write a comment...