OPEN SPECIFICATION · DRAFT FOR COMMENT

PCD-E Sandbox Ingest

How to send infusion pump and ventilator data to an open reasoning sandbox. Two doors, and the first one asks you to write no new code at all.

Version 0.2 · 26 August 2026 · Daniel Pettus · aimedagent.net/epoch

> **This specification is open, free, and unencumbered.**
>
> There is no agreement to sign, no fee, no membership, and no NDA. Anyone may implement it, fork it, or ignore it. It is built entirely on published IHE and ISO/IEEE standards and cites every one of them. Nothing in it derives from any vendor's proprietary documentation.
>
> **It is also deliberately small. If it takes your team more than an afternoon, I have specified it wrong and I would like to hear about it.**
>
> **Two version numbers, and they move independently.** This document is v0.3. The identifier on the wire, which you will see in every response as "spec": "pcd-e/0.1", is still 0.1, because no payload shape has changed. Version 0.2 added the quickstart below and the implementer notes in section 11a, both out of the first live runs. Version 0.3 removes the last human from the path: keys used to be issued by email and are now minted by the endpoint on request, which changes section 9 and the quickstart but not one byte of the schema. The wire identifier will only move when a payload shape does.

0. TRY IT IN SIXTY SECONDS, BEFORE READING ANYTHING ELSE

The endpoint is live. You do not need a key to look at it, you do not need a device to send it something, and you do not need anything from me at all.

```
# what it is and what it accepts, no key required
curl https://aimedagent.net/device

# is it up right now
curl https://aimedagent.net/device/health
```

To send something you need a key, and **you issue your own**. There is no approval step, no form, no waiting and nobody to email. Download the reference client, which has no dependencies and needs only Node 18 or newer:

```
curl -O https://aimedagent.net/pcde-client.mjs

# mints a key in your name, generates a synthetic five-minute
# pump window, sends it, and prints what came back
node pcde-client.mjs --org "Your Company" --demo
```

Or mint one directly, if you would rather point your own gateway at the endpoint:

```
curl -X POST https://aimedagent.net/device/key \
  -H 'Content-Type: application/json' \
  -d '{"organization":"Your Company"}'
```

The demo builds ten IHE PCD-01 messages over five minutes at 125 mL/h, drops the rate to 5 mL/h at 14:08, and resets the cumulative volume counter partway through. It touches no real data and needs no device. What comes back is the epoch and what a reasoning engine made of it.

> **The two lines at the bottom of that output are the entire argument.** first 125 mL/h and last 125 mL/h are what a five-minute point sample would have recorded, and both are unremarkable. min 5 mL/h at 14:08:00 is what the summary keeps. Same window, same device, four extra numbers, and one of them is an event that would otherwise not exist.

When you are ready to send your own, point the client at a file of HL7 your gateway already produces:

```
node pcde-client.mjs --key YOURKEY --hl7 your-window.hl7
```

1. WHAT THIS IS, AND WHAT IT IS NOT

The sandbox is a public endpoint that accepts periodic and event data from an infusion pump or ventilator gateway, computes a five-minute interval summary from it, hands that summary to a reasoning engine alongside a synthetic patient chart, and returns what the engine concluded.

It exists to answer one question that the field has never had a cheap way to ask: *if a reasoning system could actually see what the device did, what would it say?*

It is not

- A medical device, or a component of one. Synthetic data only, no patients, no clinical use.
- A product, a service, or anything with a business model behind it.
- A conformance test. It does not certify anything and it issues no statement about your product.
- A data store. Nothing is retained. See section 7.

> **Why the direction is inbound.** Every attempt at device interoperability for twenty-two years has asked a manufacturer to build and expose an endpoint. That ask has a perfect record of not being answered, for reasons that are economic rather than technical. This inverts it. The endpoint already exists and it is mine. You point something at it.

2. TWO DOORS

Pick whichever is less work. They are equivalent in what they produce.

	DOOR 1 · SEND WHAT YOU ALREADY SEND	DOOR 2 · SEND A JSON EPOCH
For whom	Any gateway that already implements IHE PCD-01, PCD-10, or both. That is most installed infusion connectivity built since roughly 2010.	Cloud-native devices, new products, and anyone who would rather not stand up an HL7 v2 sender for a sandbox.
What you send	`POST` the HL7 v2 message you already emit, as the request body, content type `application/hl7-v2`. Unchanged. No re-mapping.	`POST` a JSON object matching the schema in section 5, content type `application/json`.
Who computes the epoch	**We do.** Send your normal periodic cadence, whatever it is. The sandbox buckets and summarizes.	You do, or we do. Send raw samples and we summarize, or send a pre-computed summary and we pass it through.
New code on your side	**None.** A destination address and a bearer token in a header.	A JSON serializer.
Endpoint	`POST /device/hl7`	`POST /device/epoch`

Door 1 is the point of the whole exercise. If your gateway already speaks PCD-01, the total integration effort is a URL and a header.

Live now at **https://aimedagent.net/device/hl7** and **https://aimedagent.net/device/epoch**. The capability statement at https://aimedagent.net/device is unauthenticated and always reflects what is actually deployed.

3. WHAT THE SANDBOX ACCEPTS, BY TRANSACTION

All references are to the **IHE Devices Technical Framework, Revision 10.0, Final Text, 4 November 2024**, Volumes 1 to 3. The domain was renamed from Patient Care Device, which is why transaction identifiers keep the PCD prefix.

TRANSACTION	PROFILE	ACCEPTED	NOTES
PCD-01	DEC	**Yes, primary**	Communicate PCD Data. Periodic observations. This is the one the epoch is computed from. TF-2 section 3.1.
PCD-10	IPEC	**Yes**	Communicate Infusion Event Data. Final Text since Rev 9.0, 12 December 2019. Events are carried through to the context block. TF-2 section 3.10; the event enumeration is in TF-2 Appendix M.
PCD-04	ACM	**Yes**	Report Alert. Alarms bypass the epoch entirely and are timestamped on arrival. See section 6.
PCD-03	PIV	Accepted, parked	Communicate Infusion Order. Accepted and logged so a round trip can be demonstrated later. The sandbox does not program anything and never will.
PCD-02	n/a	Not applicable	Marked Reserved in TF-2 section 3.2. Note that TF-1 section 4.3 still refers to it by name. Do not build subscription on it.
PCD-05 to 07	ACM	Not accepted	Alert status and dissemination are a closed loop between an alert manager and communicators. Out of scope for an inbound sandbox.
PCD-09	IDCO	Not accepted	Implantable device cardiac observations. Different domain.

Ventilators

There is no ventilator profile in the framework. Ventilator data rides on DEC and PCD-01 generically, exactly as pump data does, and the sandbox treats it the same way. The terminology, however, *does* exist and is published: **DEV TF-3 Rev 10.0, section 7.2**, is a populated device specialization content module with a containment tree and metric tables. TF-3 carries an Editor's Note stating the work "should also ultimately result in a Device Specialization – Ventilator Integration Profile," which is IHE saying plainly that the profile does not yet exist.

4. THE EPOCH

An epoch is a fixed window of time and everything the device reported inside it, reduced to a summary. It is not a sample. The distinction is the entire reason this specification exists.

4.1 Interval

Configurable per participant, from **one minute to ten minutes**, set when your key is issued and changeable by email. Default five minutes.

Five is the default because it is not a novel number. The eICU Collaborative Research Database has archived bedside monitor data as five-minute medians for over a decade, which means anyone objecting that five minutes is too coarse is objecting to the dataset their own institution publishes on. One minute exists so that

the interval itself becomes a testable variable rather than an assumption. If the reasoning is materially better at one minute than at five, that is a finding worth having and nobody currently has it.

4.2 What a summary contains, per metric

FIELD	TYPE	MEANING
`n`	integer	Count of samples the device actually reported in the window. Not the count expected.
`min`	number	Minimum observed value.
`min_at`	timestamp	When the minimum occurred. This field is the difference between a summary and an average.
`max`	number	Maximum observed value.
`max_at`	timestamp	When the maximum occurred.
`first`	number	First value in the window.
`last`	number	Last value in the window.
`mean`	number	Arithmetic mean of observed values.
`unit`	string	UCUM. See section 8.

Eight numbers where a one-hertz stream would have produced three hundred. Roughly a fortyfold reduction in volume that increases rather than decreases what a reader can conclude, because the extremes survive with their timestamps.

> **The failure mode this is designed against.** Julian Goldman published a slide in 2016 showing a patient monitor that displayed an oxygen saturation of 84 percent at 14:07 and alarmed, next to an anesthesia record for the same period showing an unbroken line of normal saturations. His caption is two words: sampling error. A five-minute point sample reproduces that failure exactly. A five-minute summary carrying a minimum of 84 with a timestamp of 14:07 does not.

4.3 Cumulative counters reset, and the sandbox must know it

Cumulative volume delivered is not monotonic. On a real pump it restarts when the infusion is reprogrammed, when a programmed parameter other than rate or dose changes, and when an infusion is restored after completing. Any consumer that computes volume as a simple delta across an interval boundary will produce a negative number, or a badly wrong positive one, when the counter resets mid-window.

So the sandbox does not compute volume deltas. It reports **first** and **last** for cumulative metrics and sets a context flag, **`counter_reset`**, when last is less than first. Consumers of the output are told the reset happened rather than handed arithmetic that quietly absorbed it.

5. DOOR 2: THE JSON EPOCH

One object per device per interval. Send raw samples and the sandbox summarizes them, or send a summary you computed and it passes through. Both are shown.

```
POST /device/epoch
Authorization: Bearer <your key>
Content-Type: application/json

{
```

```
  "spec": "pcd-e/0.1",
  "participant": "acme-pumps",
  "device": {
    "kind": "infusion-pump",        // or "ventilator"
    "model": "Acme 4-Channel",
    "serial": "SIM-0007",           // simulator serial, never a real one
    "channel": 2                    // omit for single-channel
  },
  "patient_ref": "synthea-bca1691f", // MUST be a synthetic id. See section 7.
  "window": {
    "start": "2026-08-26T14:05:00Z",
    "end":   "2026-08-26T14:10:00Z"
  },
  "metrics": [
    { "code": "MDC_FLOW_FLUID_PUMP", "unit": "mL/h",
      "samples": [ [ "2026-08-26T14:05:00Z", 125.0 ], ... ] },

    { "code": "MDC_VOL_INFUS_ACTUAL_TOTAL", "unit": "mL",
      "summary": { "n": 300, "first": 12.4, "last": 22.8,
                   "min": 12.4, "min_at": "2026-08-26T14:05:00Z",
                   "max": 22.8, "max_at": "2026-08-26T14:09:59Z",
                   "mean": 17.6 } }
  ],
  "context": [
    { "at": "2026-08-26T14:07:12Z", "state": "kvo_transition" },
    { "at": "2026-08-26T14:07:12Z", "state": "rate_change",
      "from": 125.0, "to": 5.0, "unit": "mL/h" }
  ],
  "coverage": {
    "expected": ["MDC_FLOW_FLUID_PUMP", "MDC_VOL_INFUS_ACTUAL_TOTAL"],
    "missing":  [],
    "reason":   null
  }
}
```

Send samples or send a summary. Do not send both for the same metric; the sandbox will reject it rather than guess which one you meant.

5.1 Coverage is not optional

The `coverage` block states which metrics were expected and which arrived. A missing value must be explicit and must carry a reason: device disconnected, sensor removed, channel idle, gateway lost the link. **Silence is never returned as normality.** A reader that cannot distinguish "normal" from "not measured" will eventually average the two and hand the result to somebody as a fact.

6. THE CONTEXT BLOCK, WHICH IS THE PART THAT DOES NOT EXIST ANYWHERE ELSE

Every standard listed in section 3 will faithfully transmit a number. None of them carries the fact that explains it.

Goldman’s second slide is a non-invasive blood pressure cuff and a pulse oximeter on the same arm. The cuff inflates, perfusion to the finger stops, the oximeter reports a falling saturation, and the monitor charts a

desaturation that never happened. His caption: "Inflation metadata could identify BP artifact." The data is not wrong because the sampling rate is too low. It is uninterpretable because the context is absent.

6.1 Recognized context states

STATE	APPLIES TO	WHY IT MATTERS
`rate_change`	pump	Not carried by IPEC. Must be inferred from a stop and start pair.
`kvo_transition`	pump	Not carried by IPEC. A pump at keep-vein-open is still infusing, at a rate that has nothing to do with the order.
`secondary_to_primary`	pump	Not carried by IPEC. A secondary pauses the primary on a shared line. Treating them as concurrent double counts volume.
`counter_reset`	pump	Cumulative volume restarted inside the window. See section 4.3.
`nibp_cuff_inflating`	monitor	With `same_limb` true, explains an oximetry artifact.
`sensor_detached`	any	Distinguishes a missing value from a normal one.
`service_mode`	any	Device in calibration or service. Values are not patient data.
`patient_association_changed`	any	The framework explicitly places patient identity binding out of its own scope. It is the highest-consequence unsolved problem in this domain.
`clock_resync`	any	Event ordering before and after this point is not comparable.

> **Three of those states are the argument.** Rate change, secondary-to-primary switchover and transition to keep-vein-open are precisely the three infusion states IPEC does not carry. I verified this against the current framework: the event set in TF-2 Appendix M is Delivery Start, Delivery Stop and Delivery Complete as required, plus optional communication status change, program cleared, auto-program cleared, patient ID change, patient weight change and patient change. Nothing else.
>
> They are not unreachable. A gateway shipping in 2013 conveyed a rate change as a Delivery Stop with status standby and actual rate zero, immediately followed by a Delivery Start with status infusing. It worked. But it means the state exists only in the relationship between two messages, and a consumer that drops one half, or keys on the event code without reading mode and status together, has silently mis-recorded therapy and will never know.
>
> **The pump already knows all three. Putting them in a context block costs the gateway nothing, because it was told.**

7. SYNTHETIC DATA ONLY

> **Send simulator output. Never send data from a real patient, scrubbed or otherwise.**
>
> The word "de-identified" does not appear anywhere in this specification and must not appear in your implementation. De-identified means real patient data with identifiers removed, which is a compliance question about provenance that neither of us can audit. Data that was never about a person is not protected health information at all, because it is not individually identifiable health information under 45 CFR 164.514(a). Those are different categories carrying very different risk, and only one of them is acceptable here.

7.1 The tripwire

A request is rejected with **`400`** and is not written to disk or to any log if it carries a populated patient name, a date of birth, a patient identifier outside the published synthetic allowlist, or any field matching a medical record number, social security number, telephone or email pattern. The rejection response names the offending field path and nothing else. It does not echo the value.

This runs *before* persistence, not after, because the realistic failure mode here is not an attack. It is a competent engineer pointing the wrong stream at the URL on a Friday afternoon.

7.2 Retention

There is none. The sandbox computes, reasons, returns the result in the response, and discards the request. Nothing is stored at rest. That removes the entire category of questions about purge procedures, breach notification and data subject rights, and it is why this specification can be published without a data processing agreement attached to it.

8. TERMINOLOGY, AND AN HONEST ACCOUNT OF ITS STATE

Metric codes use ISO/IEEE 11073-10101 nomenclature as published in **IHE DEV TF-3 Rev 10.0, section 7.1 for infusion pumps and section 7.2 for ventilators**. Units are UCUM. Those two tables are free, downloadable from ihe.net, and are the only terminology source this specification depends on.

8.1 Two rules the sandbox applies to itself

- **No code is emitted from memory.** Every term the sandbox writes into its own output was looked up in a published table and carries a citation to the volume and section it came from. A term that could not be verified is marked **`UNVERIFIED`** in the output rather than guessed at, because an almost-correct MDC term is worse than a missing one: it validates.
- **Provisional identifiers are resolved, not passed through.** Terms carrying the **`MDCX_`** prefix were placeholders IHE used before IEEE assigned final codes. TF-2's own revision history carries the change item to update them to final terms from 11073-10101b, and no MDCX_ term appears anywhere in Rev 10.0. If your gateway still emits them, the sandbox will map what it can, flag what it cannot, and tell you which.

8.2 What is genuinely broken, said out loud

Rosetta Terminology Mapping is **required** by DEV TF-1 section 2.6, which states plainly that use of RTM is required in IHE-DEV profiles. It is **unwritten** in TF-3 section 4, which is an empty placeholder reading "Section reserved for future updates." The IHE wiki page describing the current RTM release was last edited in May 2011 and points at an FTP URL that no current browser can open. The live registry, NIST RTMMS, is described by NIST as freely available but serves a login page and requires registration to look anything up.

> **Required in Volume 1, unwritten in Volume 3, frozen at 2011 on the wiki, and gated behind a registration form at NIST.** That is the state of the terminology layer that every profile in this domain depends on, and it is a reasonable part of the answer to why device data has not moved in twenty-two years. This specification works around it by depending only on the TF-3 tables, which are published and free.

9. AUTHENTICATION, DELIBERATELY MINIMAL

A bearer token in an **`Authorization`** header, over TLS 1.2 or better. You mint the token yourself at **`POST /device/key`** by supplying an organization name. There is no registration form, no approval step, no client secret rotation ceremony, and no JSON Web Key Set to host.

The organization name is carried inside the key, so a connection is attributable without a lookup and no list has to be kept anywhere. That is the only job the key does. **It is a name tag, not a security boundary**, because there is nothing behind the door worth forcing: the endpoint stores nothing and only ever sees synthetic input.

One line is written to the server log for each mint and each connection: a fixed marker, whether it was a mint or a send, and the organization name you supplied. No payload, no field, no value, ever. That line exists because a participation study with no numerator is not a study, and it is disclosed here rather than buried.

Why not SMART Backend Services, which is the correct standard

Because it would require you to build something. SMART Backend Services uses client credentials with a signed JWT assertion, and the specification strongly prefers that the client publish its public keys at a TLS-protected JWKS URL. Read that requirement again from the perspective of a device manufacturer: it asks the sender to stand up and maintain a public endpoint, which is precisely the ask this entire project exists to remove.

There is a second reason and it is methodological. If the sandbox requires JWKS hosting and nobody connects, I cannot distinguish industry indifference from authentication friction, and my own result becomes uninterpretable. Every requirement added here is a confound in a study I intend to publish honestly. So the requirements are stripped to the bone, and this paragraph is part of the record that they were.

SMART Backend Services is published as an optional upgrade path and the endpoint will accept it. Nothing is gated behind it.

10. WHAT YOU GET BACK

The response is the point. This is not an inbox.

```
200 OK
{
  "epoch": { ...the normalized summary the sandbox computed... },
  "coverage": { "performed": 6, "of": 8, "missing": ["..."] },
  "reasoning": {
    "summary": "one line, under 140 characters",
    "detail":  "what the engine concluded and why, in markdown",
    "cites":   ["which values in your epoch drove the conclusion"],
    "gaps":    ["what it could not assess, and why"]
  },
  "terminology": {
    "resolved":   [ { "sent": "...", "mapped_to": "...", "source": "TF-3 7.1.2-
1" } ],
    "unverified": [ "any term the sandbox could not cite a source for" ]
  },
  "notice": "Research demonstration. Advisory only. Not a medical device.",
  "spec": "pcd-e/0.1"
}
```

The **`gaps`** and **`unverified`** arrays are not padding. A reasoning engine that does not state what it could not assess is writing confident prose over a hole, and the coverage statement is the only thing standing between a demonstration and a liability.

11. WHAT GETS PUBLISHED, AND WHAT DOES NOT

- **Published:** this specification, the endpoint uptime record, a working reference client anyone can run, and the reasoning output over synthetic data.
- **Published only with your written consent:** your participation, your name, your device, and anything that identifies your implementation. Silence is not consent. If you connect and say nothing, you are not in any published count.
- **Never published:** your payloads.

> **If you would rather not participate, that is a legitimate answer and I would value hearing why more than I would value the connection.** The most likely reason a manufacturer declines is that counsel will not approve pointing any stream, even a simulator, at an unaffiliated endpoint without a contract in place. If that is the reason, it is a more interesting finding than participation would have been, because it says the constraint on experimentation in this field is legal rather than technical. I will publish that conclusion, without your name, and it will be the fairer story.

11A. THREE THINGS THE FIRST LIVE RUNS TAUGHT, WHICH WILL SAVE YOU AN HOUR

These are not hypothetical. Each one happened on a real run against this endpoint, and each one produced output that looked entirely plausible while being wrong.

Send a full HL7 timestamp, and check what your serializer actually emits

An HL7 TS is **`YYYYMMDDHHMMSS`**. A date alone is legal and a date with an ISO **`T`** separator is common from anything that formats an ISO string and strips punctuation carelessly. Both used to arrive here as midnight, silently, and every timestamp in the epoch collapsed onto one instant while the summary continued to look correct.

The endpoint now reads the precision of every timestamp and says so. If it receives a stamp with no time of day, the response carries a note reading that those instants are not comparable and no event ordering should be read from the epoch. If every observation shares one instant it says that too. **You will get told rather than getting midnight**, but it is faster to check your serializer once than to read that note.

Put the MDC dimension term in OBX-6.2, not only the code in OBX-6.1

A unit sent as `265266` alone comes back as `265266`. Sent as `265266^MDC_DIM_MILLI_L_PER_HR^MDC` it comes back as mL/h with the dimension term alongside. The sandbox will not guess what a bare numeric dimension code means, and it says in the response that it could not.

Status and text values are collapsed into runs, deliberately

A pump status repeated at every sample is noise, and it buries the one sample that differs. Ten observations of which nine are infusing and one is keep-vein-open are returned as three runs with a transition count, not as ten rows. The keep-vein-open moment becomes a run of length one between two longer runs, which is the shape a reader can actually see.

12. OPEN QUESTIONS

1. **Interval boundaries.** Wall clock aligned, or aligned to first sample received? Wall clock makes epochs comparable across devices. First sample makes them comparable across a single infusion. Currently wall clock. Arguments welcome.
2. **Multi-channel containment.** The framework models a multi-channel pump as one system with multiple virtual devices and channels, and flattening that hierarchy loses which channel delivered which drug. On a two-channel pump running a vasopressor and a maintenance fluid, that is the entire meaning of the message. Version 0.2 carries a channel integer, which is almost certainly not enough.
3. **Dose without its basis.** A pump reports rate. An order may be in mcg/kg/min. Converting between them needs concentration and patient weight, and neither is guaranteed to be in the device message. A dose carried without the concentration and weight that produced it cannot be recomputed or audited downstream. Version 0.2 asks you to carry the basis or omit the dose.
4. **Ventilator context states.** The pump list in section 6.1 came from forty years of watching pumps. The ventilator list is thin and I would rather borrow somebody else's scars than invent them.
5. **Clock skew.** Event ordering assumes consistent timestamps across devices, and gateways drift. An epoch that says the rate changed before the infusion started is a clock problem, not a device problem. Version 0.2 records the arrival time alongside the device time and does not attempt to reconcile them.

13. SOURCES

Every standard cited here is free to download. Nothing in this specification derives from any vendor's proprietary or internal documentation.

- IHE Devices Technical Framework, Revision 10.0, Final Text, 4 November 2024, Volumes 1, 2 and 3. Transaction definitions in TF-2 section 3. Infusion pump event enumeration in TF-2 Appendix M. Device specialization content modules, including the pump and ventilator MDC tables this specification binds to, in TF-3 sections 7.1 and 7.2. RTM requirement in TF-1 section 2.6; the corresponding TF-3 section 4 is an empty placeholder. profiles.ihe.net/DEV
- IHE IPEC supplement, Rev 1.1 Trial Implementation 2011 and Rev 1.4 Trial Implementation 2014. Cited for the event enumeration and for the MDCX_ to MDC_ term transition between the two. ihe.net
- ISO/IEEE 11073-10101 nomenclature and 11073-10201 domain information model, 2020 editions. Cited by reference through the TF-3 tables. IEEE holds copyright; the codes used here were read from the IHE tables, not from the IEEE documents. ieee.org
- UCUM, Unified Code for Units of Measure. ucum.org
- NIST Rosetta Terminology Mapping Management System. Described by NIST as freely available; serves a login page and requires registration. Noted in section 8.2 as a gap, not used as a source. rtmms.nist.gov
- 45 CFR 164.514(a). The basis for synthetic-only rather than de-identified, in section 7. hhs.gov
- SMART App Launch, Backend Services, v2.2.0. Cited in section 9 for why it is offered and not required. hl7.org/fhir/smart-app-launch
- Goldman JM, "Safe Medical Device Interoperability to Enable Healthcare Transformation," MDIC webinar, 23 February 2016. Source of the sampling-error and blood-pressure-artifact cases in sections 4.2 and 6. mdic.org
- eICU Collaborative Research Database v2.0. Bedside monitor data captured as one-minute averages, archived as five-minute medians. The precedent for the default interval in section 4.1. eicu.mit.edu

Daniel Pettus spent forty years in medical device and health IT leadership at Alaris, CareFusion and BD, contributed to IHE Patient Care Device interoperability standards, and is named on two United States patents. He writes Inside the Loop at insidetheloopdp.substack.com and is the

author of The Technology Was Never the Problem, at pettusbook.com, which is about this subject and how forty years of it actually went. This specification is engineering documentation, not clinical guidance. The sandbox is a research demonstration. Advisory only. Not a medical device. Not for clinical use.