# Practical Guide: From Wearable Data Stream to Clinical Action

We already capture the data. How do we deliver information a clinician can act on?

**Our view:** a device data stream becomes clinical action only when every step between measurement and action is designed on purpose.

Teams invest in the sensor and the dashboard. The failures sit in the handoffs between them: a signal that arrives without patient context, an alert nobody configured, a result that never returns to the record.

*This guide walks one signal through six stages, with a check question and a failure sign at each. The one-page schema works as a standalone checklist.*

## The Data path

The path has six stages: measurement, transmission, context, interpretation, display and action. Each stage hands something to the next: a payload with a quality status, a delivery state, a patient and an episode, a classified event, a prioritized view, a task or a written-back report.

In the products we have built, the hard parts are these handoffs: an event that becomes a task, a report that returns to the hospital system, an alert rule someone can configure.

A raw ECG stream can be captured perfectly and still give a clinician nothing if the next stage cannot tell which patient it belongs to. The reverse also holds: a dashboard cannot repair a reading that left the device without a quality flag. The schema on the last page carries all six stages.

# Stage by stage

## 1. Measurement:

### Define the signal before interpreting it

**Decision.** Settle what the device captures, at what sampling or measurement rate, and which quality indicator travels with each reading: a signal-quality flag on a raw ECG strip, a confidence value on a derived parameter. A raw waveform, a derived value and a device-generated event need different processing, so the payload must declare which one it is.

- [ ] **Check Question.** Can a downstream rule reject a reading on quality, using data the device itself sent?

**Failure Sign.** A raw stream arrives with no quality flag, so every sample is trusted equally, including motion artifact and loose-electrode noise.

**Example.** A remote cardiac monitoring platform starts from a raw ECG stream; a monitoring product with a clinician app works from measured parameters.

**What Good Looks Like.** Every stored reading carries type, source, timestamp, rate and a quality indicator before anything consumes it.

## 2. Transmission:

### Make gaps and delays visible

**Decision.** Name the hops from device to app to cloud, and decide which component notices when an expected update does not arrive. After a dropout, does a reading arrive late, arrive twice or vanish? Without a stale-data label, a dropped connection looks the same as a quiet patient.

- [ ] **Check Question.** If a device reconnects after an outage, what does the clinician see for the missing interval?

**Failure Sign.** A gap or delay the clinician cannot see: the chart looks continuous and is not.

**Example.** The cardiac monitoring platform runs across a third-party device, an app, a cloud service and a portal, so each hop needs its own delivery state.

**What Good Looks Like.** A dropped connection delays a reading and never silently deletes it, and the gap is visible.

## 3. Context:

### Attach the signal to the right patient and episode

**Decision.** Decide which patient, session, device and episode each value belongs to at ingest, and which clinical system is authoritative for each identifier. When the context lives in a hospital system, define the required fields and what happens to a reading that does not match.

- [ ] **Check Question.** If you pull one value from the stream at random, can you name whose it is and from which device without leaving the product?

**Failure Sign.** A number with no patient or episode attached, landing in a queue where someone matches it to the right chart by hand.

**Example.** In a hospital-system integration, data comes from the hospital information system (HIS) through a secure API.

**What Good Looks Like.** Patient, episode, device and time stay attached end to end, and unmatched data has an exception route.

## 4. Interpretation:

### Define the rule behind the alert

**Decision.** Decide who defines classifications, thresholds and alert rules, and whether they live in configuration or in code. Keep three things apart: an event the device generates, a classification the application makes, and an alert rule that starts a workflow.

- [ ] **Check Question.** Can a clinical lead change a threshold or add a rule without a software release?

**Failure Sign.** Every reading is treated as equal, with no weighting and no suppression window, so alerts multiply and the team stops trusting them.

**Example.** The cardiac monitoring platform classifies events, and a configurable-alert monitoring platform lets alerts be configured.

**What Good Looks Like.** Each alert traces to a defined rule, and every configurable setting has a clinical owner.

## 5. Display:

### Show the next useful fact first

**Decision.** Decide what opens first, in what order and in which role view. A clinician mid-shift triages and does not read a stream, so the choice is between a prioritized list and a raw feed.

- [ ] **Check Question.** When a clinician opens the app at the start of a shift, does the first screen answer "what needs me now"?

**Failure Sign.** The clinician rebuilds the picture by hand, opening several screens to work out which patient needs attention first.

**Example.** The cardiac monitoring platform prioritizes events, and the monitoring product gives clinicians an app.

**What Good Looks Like.** The first screen ranks what needs attention now, with the event, its time and the patient context. The raw stream is one step away, not the default.

## 6. Action:

### Assign the next step and close the loop

**Decision.** Decide what happens after the insight appears and who owns it: a task, an alert route, a report, or a write-back to the clinical record. Decide also what the product does when nobody acknowledges the item.

- [ ] **Check Question.** Does every actionable result have a destination, an owner and a status that returns?

**Failure Sign.** The insight stays in the product, or the report is generated and never reaches the record.

**Example.** The cardiac monitoring platform creates tasks, and in the hospital-system integration a report is posted back into the HIS.

**What Good Looks Like.** Every actionable result ends as an owned task or a record entry, with a traceable completion or exception state.

# Implementation examples

## Raw ECG to prioritized event and task

A remote cardiac monitoring platform covers raw ECG, event classification, prioritization and tasks in one chain, across a third-party device, an app, a cloud service and a portal.

Read against the schema, the raw stream enters at measurement, events are classified at interpretation, prioritization shapes the display, and tasks close the loop at action.

**Inspect the same three points in your own product: where the quality status is set, where an event becomes a task, and who works the task.**

## Parameters, configurable alerts and clinician views

Two monitoring products show the middle of the path. One works from measured parameters, raises alerts and gives clinicians an app. The other lets alerts be configured and produces reports.

**Ask of any product: which parameters enter the system, who can change an alert rule, and what the clinician or the report shows as a result. Trace one alert end to end: the parameter that triggered it, the rule that raised it, the person who can edit that rule, and the view or report where the clinician sees it.**

## Hospital data in, report back

A hospital-system integration shows a closed loop in compact form: data comes from the HIS through a secure API, and a report is posted back into the HIS. Check both directions in your own path.

A successful read says nothing about whether the write-back arrives, and a report that never reaches the record leaves the insight inside the product.

**Ask what the sending system shows when a post-back fails, and who sees it.**

## Two more EHR integrations

Two further integrations are worth reading against the same schema: a real-time sync with an EHR, delivered in 1.5 months, and integrations with two different EHRs.

## Where to start

**Pick one signal a clinician already relies on, such as a raw ECG event, and trace it through the six stages. At each handoff, record the payload, the owner, the expected state and the visible failure sign.**

Mark every point where a person bridges the gap: restoring missing context, interpreting an undocumented rule or retyping a result into the record.

Then compare your path with the schema on the next page. If a handoff does not fit, we would like to hear where.

## Compare your path with us

Empeek builds the layer between devices, data systems and clinical workflows for connected-device and remote-care products. To compare a traced path with the schema, or to talk through one handoff in your own product, write to us.

# One-page schema: Data path from measurement to action

**Score each check: Yes / Partial / No**

| | DECISION | CHECK<br>yes / partial / no | FAILURE SIGN |
|---|---|---|---|
| **Measurement** | Signal type, rate, quality flag | **Does every payload carry a quality flag?** ☐ ☐ ☐ | Raw stream, no quality flag |
| **Transmission** | Device, app and cloud hops; behavior on dropout | **Can the service tell a delayed reading from a current one?** ☐ ☐ ☐ | A gap the clinician cannot see |
| **Context** | Patient, session, device, episode and timestamp | **Is every reading tied to one patient and episode?** ☐ ☐ ☐ | A number with no patient attached |
| **Interpretation** | Who owns classifications, thresholds and alert rules | **Can a clinical lead change a rule without a release?** ☐ ☐ ☐ | All readings equal; alert fatigue |
| **Display** | What each role sees first | **Can the clinician see event, time and context in one view?** ☐ ☐ ☐ | Picture rebuilt by hand |
| **Action** | Next step and owner: task, alert route or EHR write-back | **Does each result have an owner and a returned status?** ☐ ☐ ☐ | Insight stays in the product |