Description

Title

SmartStop System

Technical Specification

1. Introduction and Overview

SmartStop is a **Clinical Decision Support (CDS)** system designed to enhance infusion pump safety by providing real-time alerts and triage to clinical staff without directly controlling therapy. It integrates with the existing TraceLoop multi-channel infusion platform as an overlay that prioritizes critical pump alarms and patient events for human intervention. The system is envisioned in two embodiments: a primary **Non-Device CDS** version that delivers smartwatch notifications to the nearest nurse (requiring no FDA device clearance), and a secondary **Device-integrated** version that can issue pump commands in emergencies (requiring 510(k) regulatory clearance). This specification describes the SmartStop architecture, workflows, regulatory rationale, claims, and launch strategy for both embodiments in the context of TraceLoop's existing control and safety framework.

Key features of SmartStop include: (1) **Proximity-based Nurse Triage** – using Bluetooth to identify and notify the closest qualified clinician for a high-priority alarm, (2) **EHR-Triggered Alerts** – optional integration with hospital EHR events to trigger watch notifications (while avoiding any autonomous therapy decisions), (3) **User Acknowledgement & Audit Trail** – structured pathways for nurse acknowledgement of alerts, all logged to an immutable audit system, and (4) **Escalation and Fail-safe** – a timed escalation ladder that involves additional personnel or an automated pump pause if the alert goes unacknowledged, clearly separating the non-device alerting function from the device-control function. By design, the primary SmartStop implementation stays within the FDA's **Non-Device CDS** criteria (no patient-specific dosing logic and transparent recommendations)[1], while the secondary implementation extends into active device safety control. Figure references in this document (FIG. 1–FIG. 5) illustrate the new SmartStop features, and references to existing TraceLoop figures (FIG. 6 onward) are included to show integration with the broader system.

2. System Architecture and Components

2.1 Primary Embodiment (Non-Device CDS Architecture) – SmartStop Notification Layer

FIG. 1 – SmartStop Notification Architecture (Non-Device CDS Model): This diagram illustrates the end-to-end architecture of SmartStop's notification system, which operates as a layer on top of the TraceLoop platform without altering pump control. In FIG. 1, a central **SmartStop Server** (101) subscribes to priority event streams from the infusion pump controller and hospital systems. These events can include infusion safety thresholds (e.g. dose limits, abnormal sensor readings) detected by TraceLoop's controller or EHR-generated alerts. Upon a trigger, the server's **Alert Router** (102) determines the appropriate caregiver endpoint using a **Bluetooth Proximity Module** (103). Each nurse is equipped with a SmartStop **Smartwatch App** (104) that continuously scans for BLE beacons or tags (105) at each bedside or on each pump. The Alert Router (102) uses the proximity data (signal strength or location tags) to identify the nearest available nurse and immediately pushes a notification to that nurse's smartwatch (104) over a secure Wi-Fi/BLE connection. The notification contains the critical details (e.g. patient ID, pump channel, and alarm reason) but **does not** execute any pump action. This architecture ensures that alerts are delivered to the most relevant clinician in real time, without requiring the clinician to be pre-assigned, thereby minimizing response time. All components communicate through hospital-approved secure channels, and the SmartStop Server (101) logs each event and notification in a **CDS log database** (106) for auditing. The SmartStop Notification Architecture operates independently of the pump's control loop – it neither influences the pump settings nor modifies therapy parameters, preserving it as a non-device CDS tool.

Description: In FIG. 1, the flow (indicated by solid arrows) begins with a **Trigger Event** (e.g. an infusion pump's safety limit exceeded or a critical lab result in EHR). The event is relayed to SmartStop Server (101). The Alert Router (102) queries the Bluetooth Proximity Module (103) for nurse location data. The nearest nurse's smartwatch (104) is selected, and a notification is sent instantly. The nurse's device displays the alert along with context (e.g. "Pump 3 high pressure – Patient A. Acknowledge?"). The nurse can acknowledge on the watch, which is sent back to the server and logged. Dashed lines in FIG. 1 represent backend logging to the Audit Ledger (discussed later in FIG. 4). Importantly, FIG. 1 depicts a **Clinical Decision Support model**: the system provides information to the clinician, who then decides on the intervention (e.g. walking to the bedside to pause or adjust the pump manually). This approach aligns with FDA's Non-Device CDS guidance – SmartStop offers notifications *based on transparent rules* (such as threshold breaches) that the nurse can independently verify (the alert message includes the basis, like the value that crossed a threshold), rather than any opaque algorithmic dosing recommendation[1]. By avoiding direct control of the pump and by exposing the underlying trigger data to the user, the SmartStop notification layer remains an **exempt CDS** tool under the Cures Act criteria.

2.2 Secondary Embodiment (Device-Integrated) – SmartStop with Automated Pump Safety

The secondary embodiment builds on the same architecture but adds a **pump command interface** for emergency intervention. This version maintains all components of FIG. 1 but introduces a regulated control link to the infusion pump (or TraceLoop controller) and expanded logic in the server for automated action if humans cannot respond. Because issuing commands to a pump constitutes a medical device function, this embodiment is subject to FDA clearance (e.g. a Class II device with 510(k)). The architecture remains largely the same as FIG. 1, with the addition that the SmartStop Server (101) is also connected to the pump network via a secure API or manufacturer-provided integration module (107). However, this control link is **dormant** in normal operation and only activated under specific escalation conditions (see §4.5, FIG. 5). The system continuously monitors acknowledgment status of alerts: if an alert escalates to a predefined critical point with no clinician response, the server's **Emergency Pump Control Module** (108) can transmit a pause/stop command to the infusion pump (109). This command path is designed with failsafes – it will only execute if the scenario meets strict criteria (pre-configured rules such as "infusion continuing unchecked despite high-severity alarm for >3 minutes"). Even in this device-integrated mode, SmartStop does not perform any dose calculation or patient-specific adjustments; it simply executes a **fail-safe stop** or predefined pump action to prevent harm, akin to an automated safety interlock. All such commands and their triggers are recorded. The secondary architecture thus extends SmartStop from a passive notifier to an active safety system. It remains integrated with TraceLoop's existing safety hierarchy – for instance, any automatic pump stop by SmartStop would interface with TraceLoop's safety arbiter so that the pump's own state machine recognizes the stop (ensuring consistency with internal fault handling as seen in TraceLoop FIG. 5's safety ladder[2]). In summary, the device-integrated SmartStop shares the notification architecture of the non-device version but adds a carefully-scoped control output that escalates a notification into an automatic pump pause if clinicians are unable to intervene in time.

3. SmartStop Workflows and Figures

3.1 Bluetooth Proximity Nurse Triage Flow (FIG. 2)

FIG. 2 – Bluetooth Proximity Nurse Triage Flow: This figure is a flowchart detailing how SmartStop uses Bluetooth Low Energy (BLE) signals to route alerts to the appropriate clinician. The goal is to triage alarms based on caregiver proximity, ensuring the fastest response. **FIG. 2** is divided into swim-lane steps for the *Device*, *SmartStop Server*, and *Nurse Wearables*:

- **Step 201: Continuous BLE Scanning:** Each nurse's smartwatch or phone app periodically scans for BLE beacons associated with patients or pumps. Conversely, each bedside or pump may broadcast a unique BLE ID. The SmartStop Server aggregates these signals to maintain an updated map of which nurse is closest to which bed. This can be done by the wearables reporting signal strengths of nearby beacons to the server.
- **Step 202: Alert Trigger Detection:** When a critical alert is raised by the infusion device or EHR interface (e.g. an infusion pump alarm, a dangerously high lab value), SmartStop's logic classifies its **Severity and Priority**. Only high-severity, actionable

alerts enter the SmartStop triage workflow (to avoid alarm fatigue). The event is time-stamped and marked for escalation tracking.

- **Step 203: Nearest Nurse Identification:** The server queries its BLE proximity data for the nurse device ID with the strongest signal (or within the same room/zone) for the location of the alert. For example, if Patient X in Room 5 triggers an alert, and Nurse A's wearable is detected in Room 5 (strong BLE signal from that room's beacon), Nurse A is selected as the primary responder. Business logic can also factor in role or qualifications – e.g., only ICU RNs for ICU alarms – but proximity is a key determinant.

- **Step 204: Notification Dispatch:** The SmartStop Server sends a push notification to the selected nurse's smartwatch. The message includes the patient identifier, alert type, and an urgency indicator. For example: "**HIGH Priority:** Patient X – infusion pump high pressure alarm. Nearest nurse please check immediately." The notification demands acknowledgement or a response (e.g., a button to acknowledge or escalate).

- **Step 205: Acknowledgement Window:** The nurse has a short window (configurable, e.g. 30 seconds) to acknowledge on the wearable. A simple tap confirms receipt and intention to respond. FIG. 2 indicates this with a decision node: *"Nurse Acknowledged within 30s?"* If **Yes**, the workflow records the acknowledgment (proceed to logging in FIG. 4) and stops further escalation—Nurse A will address the alarm. If **No** (time window passes or nurse misses it), the system moves to the next step.

- **Step 206: Escalation to Next Tier:** If the nearest nurse does not acknowledge in the initial interval, SmartStop automatically escalates. The flow then identifies the next responder(s): e.g., the charge nurse or a secondary nurse in proximity. A new notification is sent ("**ESCALATION:** Primary nurse unavailable, please respond to Patient X alarm"). This step may also simultaneously alert the originally assigned nurse for that patient if they were not the nearest – providing redundancy.

- **Step 207: Iterative Escalation:** FIG. 2 shows this as a loop. The system continues escalating in tiers if acknowledgments are not received: after the charge nurse, it may alert an attending physician or a central rapid response team at a longer interval (e.g. 90 seconds, 3 minutes, etc., see FIG. 5 for final stage). Each escalation is logged. If at any point a clinician acknowledges, the loop stops and that clinician takes responsibility.

- **Step 208: Alert Resolution:** Once a nurse or other clinician acknowledges the alert on their device, they proceed to address the issue at bedside (manually pausing the pump, assessing the patient, etc.). They may also input a quick status update or resolution on the smartwatch (optional). The SmartStop Server receives the resolution or at least the acknowledgment, and FIG. 2 ends with the alert marked as **Acknowledged/Resolved** in the system.

The Bluetooth Triage Flow ensures **the closest qualified person is notified first**, rather than a fixed assignment that might be busy or distant. This dynamic approach is purely logistical and does not involve any algorithmic clinical decision – it relies on simple proximity data and predefined priority rules, keeping it within CDS scope. All decisions in this flow are transparent and based on known factors (proximity, timeouts), which the clinicians are aware of, maintaining trust and regulatory compliance.

3.2 EHR-Event Triggered Notification Pipeline (FIG. 3)

FIG. 3 – EHR-Event-Triggered Watch Notification Pipeline: This figure depicts how external clinical events from the Electronic Health Record (EHR) or other hospital systems can feed into SmartStop's notification workflow in a compliant manner. The pipeline is shown as a sequence from left (EHR system) to right (nurse's smartwatch), highlighting that even when integrated with patient data, SmartStop performs **no autonomous therapy decisions**, only event-driven alerts.

- **Source Events (301):** On the far left, FIG. 3 shows examples of EHR triggers that SmartStop can subscribe to: e.g., a **lab result** (such as a critically high potassium level reported by the lab system), a **medication order** (such as a stop-order or a hard limit on a drug that if exceeded, warrants halting an infusion), or a **vital sign trend** (such as rapid blood pressure drop flagged by patient monitor integrated with EHR). These events are generated by existing hospital systems and are labeled with patient context.
- **EHR Interface Module (302):** SmartStop includes an **EHR Interface** that listens for these events via standard protocols (HL7 or FHIR subscriptions). FIG. 3 shows the EHR Interface (302) filtering incoming events against SmartStop's alert criteria. Only specific, high-priority events configured in the SmartStop rule set will trigger notifications. For example, an **EHR hard stop order** for an infusion (like an allergy or interaction alert entered by a physician) could be one trigger; or a lab value far outside normal range for a patient on a certain infusion might be another.
- **Rule Evaluation (303):** Each incoming event is evaluated by a simple rule engine. This engine checks: (a) Does the event type and value meet the threshold for notifying? (b) Is the patient currently on a therapy relevant to this event? For instance, if a high potassium lab comes but the patient is not on any infusion that affects potassium, SmartStop might ignore it. But if the patient is on a potassium infusion or a medication that can cause hyperkalemia, the rule will trigger an alert. **Importantly, these rules do *not* contain any dosing algorithms or patient-specific predictive logic – they are straightforward conditional checks** (if X happens, notify). This ensures compliance with CDS exemption by not performing "black-box" analysis – it's simply relaying existing clinical decision support alerts from the EHR in a more actionable format.
- **Notification Construction (304):** For an event that passes the rules, SmartStop constructs a notification payload. This includes the key data from the EHR event (e.g., "K^+ = 6.5 mmol/L" or "Stop order: Medication Y due to interaction") so that the nurse can **independently understand the reason**. Providing this basis satisfies the transparency requirement for non-device CDS[3] – the nurse sees the exact data that triggered the alert. The notification text and urgency level are set accordingly.
- **Dispatch via SmartStop Server (305):** The alert is then dispatched through the same routing mechanism as in FIG. 2. SmartStop checks which patient/bed this pertains to and invokes the Proximity Triage to find the nurse. In FIG. 3, this step passes the event into

the Alert Router (as in FIG. 1's component 102) which then delivers to the nurse's smartwatch (again indicated as 104). This step might involve formatting the message for the wearable app and ensuring the notification is prominent (e.g., high priority alerts might override Do Not Disturb on the device).

- **Wearable Alert & Acknowledgement (306):** The nurse's smartwatch buzzes and displays, for example: "**Critical Lab Alert – Patient X:** K^+ = 6.5 mmol/L (high). Check infusion and patient." The nurse acknowledges on the watch. The acknowledgement (with timestamp) is sent back to SmartStop server and logged.

- **Clinical Follow-up (307):** Though not an automated part of SmartStop, FIG. 3 notes that the nurse's action would be to follow hospital protocol – e.g., stop the potassium-containing infusion if running, notify the physician, etc., as prompted by existing clinical guidelines. SmartStop's role was just to rapidly convey the EHR alert to the right person.

- **Logging (308):** All steps are recorded: the incoming event ID, rule triggered, notification sent, and user acknowledgement. This flows into the audit trail (to be detailed in FIG. 4).

The EHR-triggered pipeline in FIG. 3 demonstrates SmartStop's extensibility – it can incorporate a wide range of hospital alerts (not just device alarms) into its nurse notification system *without* stepping into autonomous decision-making. The **key compliance point** is that SmartStop does not generate new therapy instructions or interpret the data in a proprietary way; it relies on either preset thresholds or external clinician-entered rules. Thus, even with EHR integration, SmartStop remains a **CDS tool providing information** rather than a device function. This keeps it in line with FDA's guidance that decision support software which merely provides *recommendations or alerts alongside the supporting information* for a provider (and doesn't control a device) can be considered Non-Device CDS[1]. Of course, if the secondary embodiment is enabled, certain EHR events (like a "hard stop" order) could *optionally* trigger an automatic pump pause via the device interface, but only with regulatory approval – that crossover is discussed under FIG. 5.

3.3 User Acknowledgement Pathways and Audit Trail (FIG. 4)

FIG. 4 – User Acknowledgement Pathways and Audit Trail (CDS only): This figure focuses on how alerts are acknowledged by users and how all interactions are logged. Ensuring that clinicians can **confirm and respond** to alerts easily, and that the system keeps a tamper-proof record, is critical for both patient safety and regulatory auditing (especially since SmartStop is a safety layer on top of an autonomous system). FIG. 4 illustrates two main aspects: the *acknowledgement loop* for the caregiver and the *audit trail integration*. The embodiment here is the Non-Device CDS scenario, meaning FIG. 4 does not include any pump override commands, only acknowledgement.

- **Alert Display and Acknowledge (401):** When a SmartStop notification arrives on the nurse's smartwatch (or other device), the interface provides an "Acknowledge" (and possibly "Escalate" or "Dismiss") option. FIG. 4 shows a nurse (wearable icon) receiving an alert and tapping "Acknowledge" within the app. In some configurations, the nurse might also be able to choose a quick response like "On my way" or mark the alert as false if appropriate. The **Acknowledgement Signal** (402) is sent from the device to the

SmartStop Server confirming that a human is taking charge of the situation. This stops the escalation timers for that alert. In TraceLoop's existing override system, a similar concept exists where a clinician can silence or override an alarm for a period[4][5]. SmartStop's acknowledge is analogous but purely notifies and logs, since in the non-device mode it doesn't silence the pump alarm itself (the nurse would still do that manually at the pump).

- **Alternate Acknowledgement Pathways:** FIG. 4 also depicts alternative ways an alert might be acknowledged, ensuring flexibility in hectic clinical environments. For example, if the nurse reaches the bedside and presses a **physical button on the pump or bedside device** to silence the alarm, that action could be fed back to SmartStop via the pump's connectivity (if available) as an implicit acknowledgement. Another pathway: a central **Nurse Console** (403) monitored by charge nurses could allow manual acknowledgement or re-routing of alerts (for instance, if the nearest nurse is busy, the charge nurse could acknowledge on their behalf and assign someone else). All these pathways converge into the SmartStop logging mechanism.
- **Audit Logging (404):** Every alert event and user action goes into the **Audit Trail**. SmartStop leverages TraceLoop's secure audit logging infrastructure, which was designed for an immutable record of autonomous actions[6]. In SmartStop's case, entries like "ALERT RAISED (Patient X, event Y, severity), 14:32:05", "ALERT SENT to Nurse A (device ID...), 14:32:07", "ACKNOWLEDGED by Nurse A, 14:32:20" are all recorded. FIG. 4 shows the Audit Trail as a sequence of signed log entries (perhaps utilizing a blockchain ledger or append-only log). This is labeled as **Hash-Chain Ledger (905)** in alignment with TraceLoop's FIG. 9 concept, where each event is hashed and linked[6]. The benefit is that every SmartStop interaction has non-repudiation – one can later prove what happened and when. The audit log can be queried for reporting (e.g., how many alerts were escalated, response times) and for compliance review.
- **Audit Trail Integration:** SmartStop's logs integrate with the overall system audit. As shown in FIG. 4, the SmartStop Server forwards its log events to the same **Regulatory Node** or compliance database used by TraceLoop (for example, TraceLoop's FIG. 9 includes a regulatory reporting node[7]). This ensures that SmartStop's interventions (or lack thereof) are included in the patient's treatment record. In practice, each acknowledgment might generate an entry like an OVERRIDE_REQUEST/GRANT in TraceLoop terms, though here it's not an override of automation but a response to an alert.
- **No Patient Data Stored on Device:** FIG. 4 also notes a security design: the smartwatch or phone does not permanently store patient information or logs. It acts as a transient display. The authoritative log resides on the server and hospital systems. This protects patient privacy and maintains data integrity.
- **Alarm Fatigue Mitigation:** SmartStop's acknowledgement mechanism is also carefully designed to reduce alarm fatigue. By requiring positive acknowledgement on the wearable, it ensures the nurse is aware, in contrast to passive alarm systems that might be missed. Furthermore, by logging the acknowledgements, hospitals can track if certain alerts are frequently ignored and adjust thresholds or training accordingly. The escalation

ladder (FIG. 5) also only kicks in for **unacknowledged** high-severity alerts, meaning if a nurse quickly acknowledges a less critical alert, it won't page everyone – thus avoiding excessive disruption[5].

In summary, **FIG. 4** emphasizes that **every SmartStop alert creates an audit trail and requires a deliberate user interaction to be cleared**. This closed-loop confirmation (alert → acknowledge → log) improves accountability. The audit trail, leveraging technologies like Hyperledger Fabric or secure hash chains as used in TraceLoop[7], guarantees that the sequence of events (alert generation, dispatch, acknowledgement, escalation, resolution) is recorded in an **immutable, time-sequenced log**. This not only aids internal quality improvement but also is crucial for regulatory audits or investigations (for instance, proving that an alert was delivered and accepted by a nurse within X seconds during an adverse event review).

3.4 Pump Command Escalation Path (FIG. 5 – Device-Escalation Embodiment)

FIG. 5 – Pump Command Escalation Path (Device 510(k) Embodiment): This final foundational figure outlines the sequence of steps and decision points in the scenario where SmartStop goes beyond notification and actually issues a command to the infusion pump. This figure represents the **secondary embodiment** and highlights the additional safety checks and regulatory considerations inherent in automating pump control. The flow in FIG. 5 can be seen as an extension of FIG. 2's escalation ladder, adding a last resort: an automatic pump **"Smart Stop"**.

- **Initial Alert and Nurse Notification:** The left side of FIG. 5 mirrors the earlier steps: an alarm triggers, the nearest nurse is notified (501), and the system waits for acknowledgment (as in FIG. 2). The figure quickly moves through the first two escalation tiers: after no response from the primary nurse (within 30 seconds, for example), it escalates to backup staff (502) – e.g., charge nurse at 90 seconds, attending physician or code team at 3 minutes[8][9]. These times and roles are configurable but here illustrate a typical ladder[8]. Each escalation is an opportunity for a human to intervene. Up until this point, SmartStop still hasn't controlled the pump; it's only sending alerts, consistent with the non-device mode.

- **Automated Command Criteria Check:** The critical addition in FIG. 5 is a decision node after the final escalation tier (503) that asks: *"Has a clinician responded?"* If yes, and someone is addressing the alarm, then no automated action is taken by SmartStop (the human is in the loop and will handle the pump). But if **no acknowledgement or intervention** has occurred by the final time threshold – which implies a potentially dangerous situation (e.g., the patient is receiving a medication that should have been stopped), then SmartStop's logic proceeds to an **Automated Safety Command** path. Before executing, however, an extra safety check (504) is shown: SmartStop verifies that the condition still warrants stopping the pump and that no contradictory data has resolved the issue. For instance, if the alert was a predicted overdose situation (like infusion running too fast)[10] and by this time the pump's internal safety might have already paused due to its own watchdog, SmartStop will not double-actuate. It might poll the pump status: if the pump is *still running and in alarm*, and no nurse input, then criteria for emergency stop are met.

- **Command Execution (Pump Stop):** In block (505), SmartStop issues a **Pump Stop Command** through the pump's control interface. This could be a command to pause the infusion, set rate to 0, or power down the pump's channel – essentially whatever is the safest immediate action to halt further drug delivery. This message is sent via an API provided by the pump manufacturer or an interoperable interface. Because this is a medical device action, it would be carefully implemented: the command might include the reason code and would expect an acknowledgment from the pump. In FIG. 5, once the pump confirms it has stopped, SmartStop will generate a high-priority system alert (likely to all relevant staff): e.g. "**AUTOMATED PUMP STOP executed for Patient X – Medication Y paused due to no response to critical alarm**." This transparency is important so that when a nurse or physician arrives, they know the system intervened. (This is analogous to an autonomous safety cut-off, but one that everyone is made aware of immediately.)

- **Audit and Notification of Stop:** The automated action is logged just as any user action would be. FIG. 5 shows an entry being made in the audit log: "Pump Stop Command issued by SmartStop at 14:35:00, no ack from staff." This complements TraceLoop's internal logs which might also note that an override occurred. The system may also create a specific alert notification to a broader team (e.g., hospital central monitoring) that SmartStop halted a pump. This dual notification ensures no ambiguity – it prevents a scenario where a pump stops and staff don't know why. The audit trail ties the stop command back to the triggering condition and lack of human response.

- **Post-Stop Human Intervention:** After an automated stop, clinicians would still need to assess and restart or adjust therapy manually. SmartStop would not restart the pump on its own; it only stops it and leaves further action to humans (who might decide to resume at a lower rate or switch medications, etc.). FIG. 5 might illustrate that the **clinician must manually clear the condition** on the pump before it can be restarted (for safety). This could involve requiring a credentialed user to confirm on the pump or in the SmartStop interface that it's safe to resume.

- **Failsafe Integration:** The Pump Command Escalation is effectively a **failsafe layer**. In TraceLoop's architecture, multiple failsafes exist (e.g., hardware watchdogs, safety state machines[11][12]). SmartStop's automated stop can be seen as an **L5 manual override surrogate** if no human is available[2] – essentially the system acting in lieu of the clinician to enforce a vital safety limit. It's important that this action is consistent with the system's design: for example, TraceLoop's conflict graph or safety arbiter would accept an external "STOP" as highest priority, akin to a vital override (L-0 or L-1 layer in TraceLoop terms[13]). The SmartStop command would interface at that high priority to immediately halt actuation[13], ensuring no lower-level optimization tries to continue dosing. This alignment and clarity would be part of the 510(k) submission, demonstrating that SmartStop's command fits into the existing safety hierarchy.

- **Regulatory Note:** Because FIG. 5's scenario crosses into device function, regulatory requirements dictate extensive validation. SmartStop's automated decision to stop a pump must be **deterministic and testable**. The logic would likely be considered a Class II device feature (infusion pump accessory with safety interlock function). The design would incorporate redundancy (perhaps requiring two independent conditions or a dual-signature algorithm before stopping, to avoid false positives) and the ability for clinicians to easily override if needed (though practically, overriding a stop means restarting the pump if deemed safe). The audit logs and notifications ensure that if such an automated stop is ever litigated or reviewed, there's clear provenance of why it occurred[14] (e.g., "no nurse response, pressure alarm threshold X for Y seconds"). This helps in demonstrating safety and effectiveness to regulators.

In summary, **FIG. 5** provides a clear separation between the **Non-Device CDS workflow** (which ends at notifying humans) and the **Device-Escalation workflow** (which adds an automated pump stop after all human tiers fail to respond). The figure underscores that the automatic pump command is a **last resort** – an escalation path that would only be used in dire circumstances to prevent patient harm. The existence of this path encourages that SmartStop in non-device mode should be highly effective at engaging humans (which is why the primary design focus is on quick notification and acknowledgment). Only if those measures fail does the system tip into device mode. This dual approach allows initial deployment as a low-risk, unregulated notification tool, with an upgrade path to a regulated safety feature once appropriate approvals are obtained.

4. Regulatory and Compliance Considerations

4.1 Non-Device CDS Exemption Justification

The primary SmartStop system is intentionally designed to meet the FDA's **Non-Device CDS** criteria, which exclude certain decision support software from the definition of a regulated medical device. In particular, SmartStop does **not** analyze or output any patient-specific treatment recommendations beyond those that a clinician can interpret with the raw data provided[1]. It simply aggregates triggers from medical devices and records, then delivers alerts with context (e.g., values, thresholds) that the clinician uses to make a decision. The four criteria of the 21st Century Cures Act and FDA CDS guidance can be mapped to SmartStop as follows:

- *Criterion 1:* The software does not acquire or process signals from medical devices in a diagnostic sense – SmartStop might read an alarm status from a pump or a value from the EHR, but it is not performing analysis on a waveform or radiological image. It avoids complex "black box" signal interpretations.
- *Criterion 2:* It analyzes and displays medical information (e.g., pump alarm, lab result) to the user. This is exactly its function – to show pertinent medical info on a watch in a timely manner.
- *Criterion 3:* It is intended to support a recommendation about care (essentially "attend to this patient now" is the recommendation) to an HCP, which is true – the user is a nurse or doctor receiving info to take action.
- *Criterion 4:* Most importantly, it provides sufficient information about the basis for the alert so the HCP need not rely solely on SmartStop. For example, instead of just saying "Check Patient X", it will say "High BP alarm on Patient X's vasopressor pump" or "K^+=6.5 on Patient Y – consider stopping infusion". The nurse can see the patient's data (blood pressure value, drug name, etc.) and use their own judgment. This transparency is

key to exemption[3]. The clinicians can always verify the alert by looking at the pump or EHR themselves; SmartStop is simply expediting their awareness.

Additionally, SmartStop's algorithms are straightforward and based on hospital-defined rules. It does not adapt or change its behavior per individual patient beyond those set points (no machine learning that would create opaque logic). This further strengthens its position as an *assistive tool rather than an autonomous device*. During development, the team will document how each alert rule is derived from clinical protocols or manufacturer's device settings – nothing is purely SmartStop's own "decision." This documentation can be used to confirm Non-Device CDS status or, if needed, to support a modest Class I device classification argument (like an alarm communicator) if regulators deem any part of it under device oversight.

4.2 FDA Oversight for Device-Integrated Mode

The secondary embodiment, where SmartStop sends a pump stop command, is clearly a device function and will be treated as such. This mode effectively makes SmartStop a combination of a secondary alarm system and an **infusion pump accessory with automatic shutoff capability**. Likely, this falls under infusion pump safety interlocks, which FDA would regulate. The plan for this embodiment is to pursue a 510(k) clearance, possibly under an existing product code for infusion pump monitors or alarm systems. We anticipate the need for demonstrating substantial equivalence to something like a remote pump stop or an ICU patient safety system.

Key regulatory points for this mode include: fail-safe design, validation of the algorithm (must show it reliably triggers only when intended and improves safety), cybersecurity for the command link, and human factors (ensuring users understand when the system will act and how to respond). SmartStop will log every device command and its reason, providing the traceability required under 21 CFR 820 (Design Controls and Device History) in the audit trail[14]. The device mode will not be activated without regulatory approval; in fact, it may be distributed as a separate "SmartStop Pro" module or a firmware update that is disabled until clearance, so that the base notification functionality can be used freely.

4.3 Compliance with Alarm Standards and Integration

SmartStop's alerting logic will comply with existing medical alarm management standards (e.g., IEC 60601-1-8 for alarm systems in medical equipment). Although SmartStop itself is software, by dealing with alarms, it respects priority classifications (like distinguishing a high priority physiological alarm vs low priority technical alarm and not overwriting their meaning). It will complement the pump's alarms, not replace them: the pump will still emit its audible/visual alarm as per its design; SmartStop simply provides an additional communication channel to ensure it's acted upon. For the device version, any command it issues (like stopping an infusion) will follow the pump's communication protocol and safety requirements, possibly using the pump's manufacturer-provided API or integration service to ensure compatibility and no unintended consequences.

4.4 Optional EHR Integration Scope and Compliance

As described in FIG. 3, SmartStop's integration with the EHR is kept to **event triggers only**. It does not write any data to the EHR (except possibly documentation notes that an alert was acknowledged, if desired, but that would be akin to writing a log or message). It reads certain flags or results and triggers alerts accordingly. This means it stays on the **right side of interoperability**: using standard HL7/FHIR inputs and not altering the source of truth. From a compliance perspective, we ensure that any EHR data used is protected (SmartStop will be deployed on hospital servers or cloud with proper HIPAA compliance, etc., as it handles PHI in alerts). The integration will be configured so that it *only subscribes to necessary data* – e.g., specific lab thresholds relevant to therapies the TraceLoop system is managing, or specific medication orders. This minimizes unnecessary data flow and reduces risk of information overload or privacy exposure.

Because SmartStop doesn't make decisions, it also doesn't violate the FDA's stance that if software is directly treating or diagnosing based on patient-specific information, it becomes a device. Instead, it treats EHR alerts much like a human would treat a clinical decision support alert: something to act on if relevant. If the hospital already vetted those alerts (e.g., drug-drug interaction warnings, or sepsis alerts), SmartStop is just a delivery mechanism to the nurse's wrist.

4.5 Data Security and Audit Trails

Given SmartStop's role in patient safety, the system implements robust security and audit measures. All communications between the SmartStop server, pumps, EHR, and wearables are encrypted (TLS/SSL for IP traffic, BLE encryption for wearable links). The **audit trail** entries described in FIG. 4 are stored in an immutable format – for example, using a blockchain ledger or an append-only log with cryptographic hash chaining[7]. This ensures that no one can retrospectively alter logs (a critical point for both FDA and Joint Commission audits in case of adverse events). Each alert and action is time-synchronized (potentially using NTP or the hospital's time server) to correlate with other system logs.

SmartStop will also adhere to access control: only authorized devices (registered nurse watches, etc.) can receive alerts, and only authenticated users can acknowledge them. The system may integrate with hospital single sign-on or badge systems so that we know which nurse (ID) acknowledged the alert – this ties into TraceLoop's existing Role-Based Access Token Service concept[15], where each user's role and identity are known. Indeed, TraceLoop's FIG. 10/11 includes a badge reader and role mapping[16][17]; SmartStop can leverage the same so that, for example, only a credentialed critical care nurse's acknowledgement will halt escalation for an ICU drug alert.

5. Branding, Distribution, and Commercial Launch Strategy

Under the brand name **SmartStop™**, the system will be positioned as a next-generation infusion safety overlay. The branding emphasizes **"smart intervention stopping harm before it happens"** – aligning with the idea that the system proactively stops potential infusion-related harm (either by getting a human there in time or by pausing the pump). SmartStop will be marketed not as a modification to infusion pumps, but as a **standalone clinical support solution** that can work with existing pumps and hospital IT infrastructure. This is strategic for two reasons: regulatory agility and market penetration.

Manufacturer Partnership: We plan to partner with major infusion pump manufacturers to integrate SmartStop's connectivity. Initially, this means ensuring SmartStop can receive pump alarm data (read-only) and patient infusion parameters via standard pump networking (many modern "smart pumps" have integration capabilities or alarm forwarding to nurse call systems). A partnership could involve co-marketing SmartStop as a value-added solution that works with Manufacturer X's pumps to reduce alarm response times. Importantly, because the base SmartStop doesn't send commands to the pump, it **does not alter the pump's FDA classification or require changes to the pump's cleared indications**. It would be akin to a software accessory that the pump company can offer to hospitals to improve workflow, without needing to include it in their regulatory submission. This lowers the barrier for pump vendors to collaborate. For the secondary (command) mode, the partnership is even more crucial – any remote stop command likely needs a built-in capability in the pump's firmware or an external interface device. We would work with a manufacturer to pilot that feature, likely as a controlled upgrade in a post-market study, then jointly approach FDA if needed for a clearance covering the integrated system. The manufacturer benefits by touting enhanced safety (with SmartStop) without having to develop the feature entirely in-house.

Non-Pump Classification & Distribution: SmartStop's primary version can be distributed as a **Software-as-a-Service (SaaS)** or on-premises server in the hospital. Since it doesn't control devices, it can be seen as a hospital IT system or a clinical communication tool rather than a regulated medical device. This means we could deploy it widely while labeling it as a clinical support tool (not for making treatment decisions, only for facilitating communication). The initial target market is high-acuity settings like ICU, OR, and ED where alarms are frequent and critical. The system would be sold either directly to hospitals or through pump vendors as noted. Since it interfaces with pumps, one strategy is to **bundle SmartStop with pump sales**: e.g., a hospital purchasing new infusion pumps gets an option to add SmartStop to their purchase for improved alarm management. Alternatively, a **standalone sale** to hospitals is possible, highlighting interoperability with whatever pump systems they have (provided we support the protocols or a middleware that captures pump alerts).

Commercial Launch Phases: The launch is envisioned in phases:

- *Phase 1:* **SmartStop-Notify (CDS version)** – The initial release focusing on notifications and triage only. Marketing will focus on immediate benefits: reducing response times, reducing alarm fatigue by targeting the right person, and improved compliance (due to the audit trail of alarm acknowledgements). We will gather real-world data here to demonstrate outcomes (e.g., a study might show a 50% reduction in time to respond to critical infusion alarms when SmartStop is used). This data not only helps sales but also will be part of the safety case for the next phase.

- *Phase 2:* **SmartStop-Command (Regulated version)** – After establishing a footprint and proving the concept, we will introduce the automated pump stop feature. This will be branded possibly as "SmartStop AutoSafe" or similar, to indicate an added capability. Because this requires regulatory clearance, the timing will depend on successful FDA discussion. It might launch initially as a limited release or in specific jurisdictions. We will emphasize that this version provides a safety net beyond human limitations, potentially preventing serious adverse events (for instance, "SmartStop AutoSafe halted an overdose infusion when staff were unable to respond in time, averting patient harm" –

this kind of case study can be powerful). Pump partners might incorporate this as a firmware update or an add-on module to their pumps, sold as a premium safety feature to hospitals.

Distribution Channels: In addition to pump manufacturer channels, SmartStop will engage with **hospital IT and patient safety departments** directly. The system spans biomedical engineering (because of pump integration) and clinical IT, so we will attend industry conferences (HIMSS for IT, AAMI or APSF for patient safety, etc.) to showcase the technology. Early adopters might be large academic medical centers known for innovation in patient safety – their endorsements would be valuable. We might also seek a partnership or endorsement from nursing organizations, since SmartStop heavily influences nursing workflow (if nurses champion it as a tool that helps them rather than annoys them, that's crucial for uptake).

Pricing and Classification: SmartStop-Notify being non-device could be sold as a subscription or license per hospital unit or per pump monitored (e.g., per bed license). Because it's not a regulated device, the procurement can be quicker (just software license, like buying a nurse call system upgrade). SmartStop-Command, being a medical device software, would likely be priced higher or packaged differently (maybe a per-pump add-on cost) since it provides direct patient safety intervention.

Brand Differentiation: We will brand SmartStop not as just another alarm or pump feature, but as a **care coordination platform**. By using terms like "clinical decision support" and "care team mobilization," we position it as a modern solution for alarm fatigue and rapid response – a known pain point in hospitals[19][20]. The branding will stress: *SmartStop doesn't replace the nurse or doctor – it makes sure the right clinician is in the right place at the right time, and if all else fails, it prevents harm automatically.* This messaging addresses any fear of automation by highlighting the support aspect.

Regulatory Path and Launch Timing: We intend to launch the Non-Device CDS version as soon as validation in pilot sites is complete, leveraging the regulatory exemption to start generating revenue and real-world evidence. Meanwhile, in parallel, we will work on the regulatory submission for the device-integrated features, likely using the De Novo or 510(k) pathways by comparing SmartStop's safety intervention to existing "remote alarm pause" functions or safety interlocks. If clearance is obtained, a software update could activate those features in existing installations, meaning hospitals who already adopted SmartStop could seamlessly gain the new capability (subject to agreeing to whatever updated regulatory labeling, etc.). This is a powerful incentive for early adopters – they can be assured that they are "future-proofed" for advanced functionality once it's approved.

Summary: The SmartStop system will be introduced as a flexible, **vendor-neutral** safety layer that can work with multiple infusion pump brands and EHR systems, branded with a focus on smart safety and quick response. Its initial non-device classification allows broad and fast deployment without regulatory delay, positioning it as a novel solution for alarm fatigue and rapid response coordination. Then, via close collaboration with pump manufacturers and positive results from the field, SmartStop will evolve into a full-fledged device-enhanced safety system, distinguishing itself in the market as not just a monitoring tool but an **active guardian for infusion therapy**. By strategically straddling the line between software and device and moving that line when ready, SmartStop aims to save lives and improve workflow, all while aligning with regulatory frameworks and hospital operational needs.

References: The integration with the TraceLoop system ensures that SmartStop's functionality complements the existing closed-loop control and safety features. TraceLoop's multi-tier safety architecture (FIG. 5, FIG. 7 in prior documentation) already envisions a role for human overrides and supervision[2], which SmartStop fulfils by connecting to the clinicians in real time. The escalation ladder concept described in TraceLoop's FIG. 10[8] is effectively implemented through SmartStop's notification tiers, and the audit logging aligns with TraceLoop's FIG. 9 immutable log design[6]. By referencing these, we ensure SmartStop is not an isolated add-on, but an integrated component of a comprehensive intelligent infusion ecosystem. Each figure (1–5) in this specification builds on that foundation, and additional figures from 6 onward in the TraceLoop documentation (such as sensor-actuator matrices, rule execution priorities, watchdog timers, etc.) continue to provide the technical context in which SmartStop operates as the human-facing complement to TraceLoop's automated engine.

Overall, SmartStop's technical specification defines a system that is **innovative yet implementable**, leveraging existing technologies (BLE, smart devices, secure logs) in a new configuration that addresses a critical gap in infusion therapy management. It stands to significantly improve patient safety by uniting the speed of automation with the judgment of clinicians, under a framework that is both **regulatorily compliant and clinically practical**.

[2][6][5][1]

---

[1] [3] The Incredible Shrinking Exemption: FDA Final CDS Guidance Would Significantly Narrow the Scope of Exempt Clinical Decision Support Software Under the Cures Act

https://www.thefdalawblog.com/2022/10/the-incredible-shrinking-exemption-fda-final-cds-guidance-would-significantly-narrow-the-scope-of-exempt-clinical-decision-support-software-under-the-cures-act/

[2] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] TRACELOOP OO NEW.docx

file://file-KYjTKHC5fQnnLSfNS42yUo

[19] Quantifying the Impact of Infusion Alerts and Alarms on Nursing ...

https://pmc.ncbi.nlm.nih.gov/articles/PMC8245209/

[20] Optimizing Smart Pump Technology by Increasing Critical Safety ...

https://pmc.ncbi.nlm.nih.gov/articles/PMC4336013/