Figures

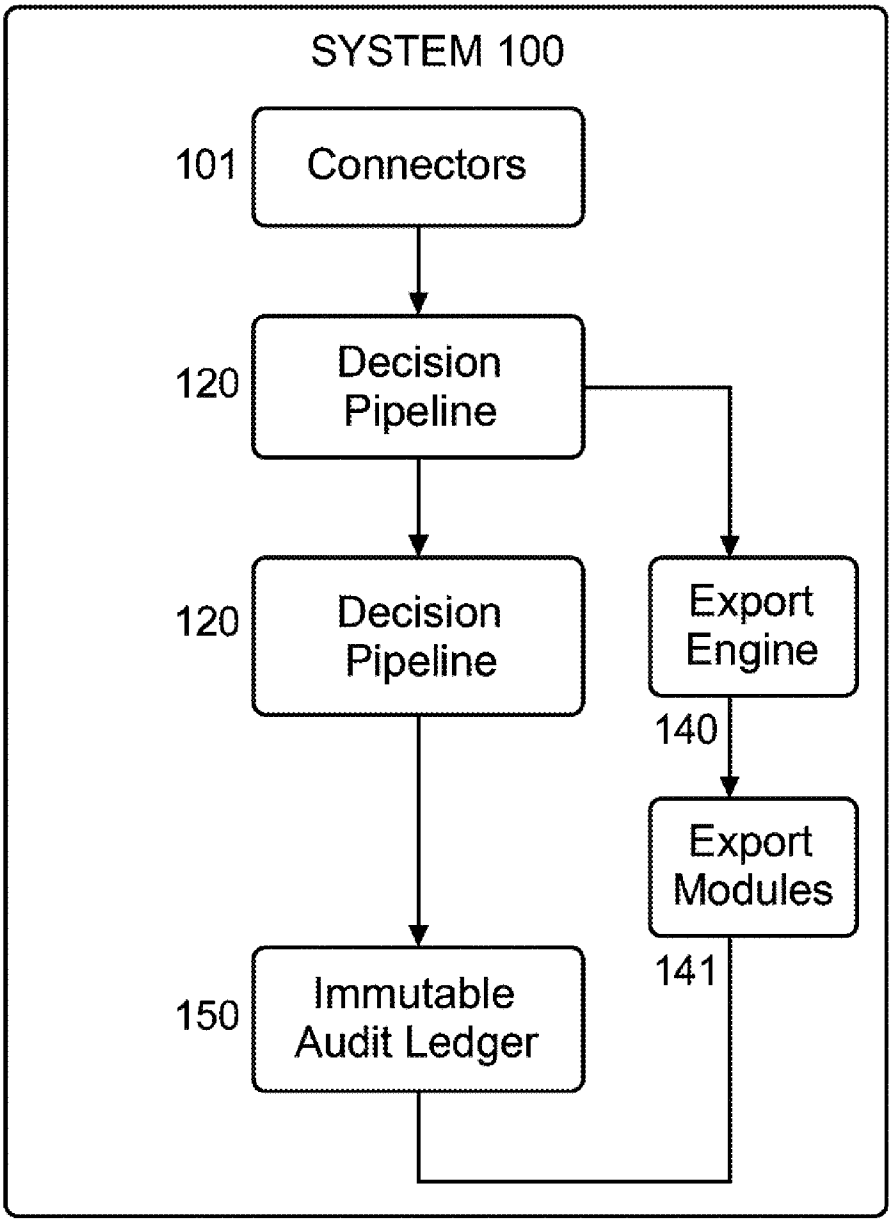

FIG. 1 – System Architecture

Figure 1:
*High-Level Architecture of **SmartStop Sentinel Connect™**, showing the end-to-end*

workflow. At the **input layer**, multiple prebuilt **connectors (101)** interface with diverse hospital systems and devices (e.g. smart infusion pumps, EHR modules, surgical trackers) to continuously ingest event data. All incoming signals feed into a central **Event Bus (110)**, where they are normalized to a unified schema for consistency. The normalized events then enter the **decision pipeline (120)** – a multi-stage processing core. The pipeline's outputs branch into two parallel paths: an **Export Engine (140)** that formats and routes reports to external systems, using various **export modules (141)** for state reporting, PSO submission, etc.; and an **Immutable Audit Ledger (150)** that records every event and decision in a tamper-proof log for compliance and traceability. This architecture is modular and layered: connectors feed the event bus; the pipeline (detailed in Fig. 2) analyzes events; the export engine outputs to required channels; and the ledger underpins the whole system with an audit trail.

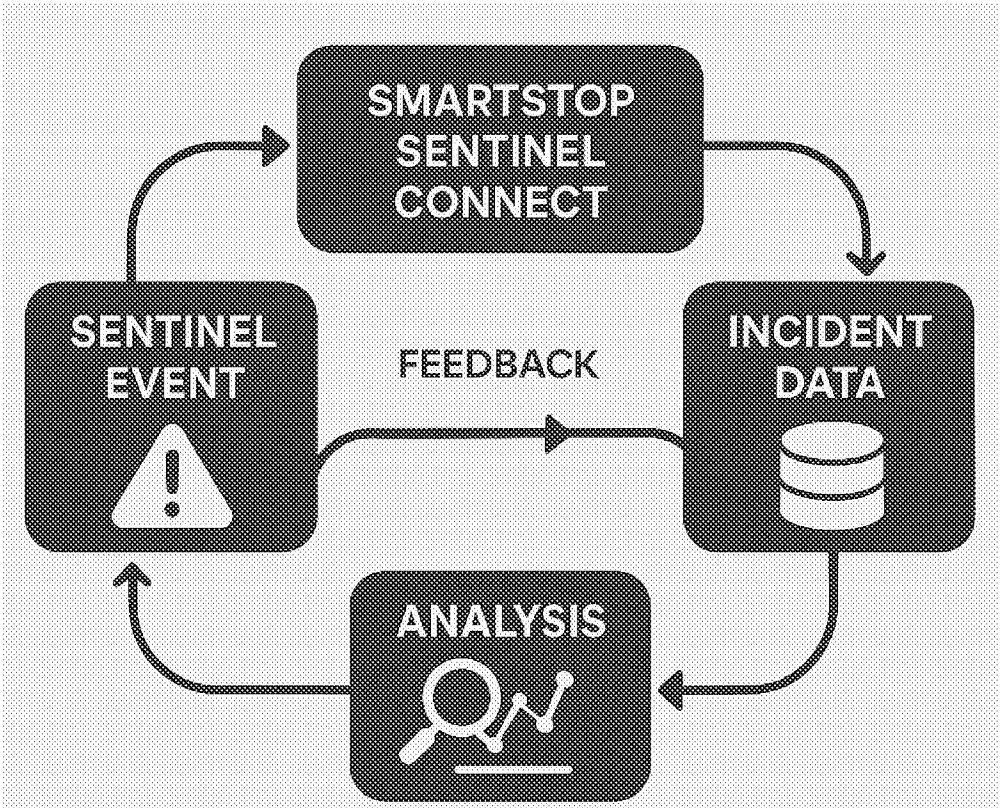

Fig. 1B
Closed-Loop Learning Cycle

Figure 1B: Closed-Loop Learning Cycle, depicting how the system enables continuous improvement. After each sentinel event is detected and managed (left side of loop), the incident data is **recorded and aggregated** in the central repository/PSO database. Those data are then **analyzed** (e.g. via Root Cause Analysis or trend analysis) to extract insights and identify safety improvements. Findings lead to **system and process enhancements** – for example, refining triage algorithms, updating rules, or adding new device connectors. These **improvements (feedback)** are fed back into the SmartStop Sentinel Connect system, which in turn strengthens future event detection and prevention. This closed-loop cycle ensures that as more events occur, the system *learns* and becomes more effective over time, continually increasing the hospital's ability to prevent harm.

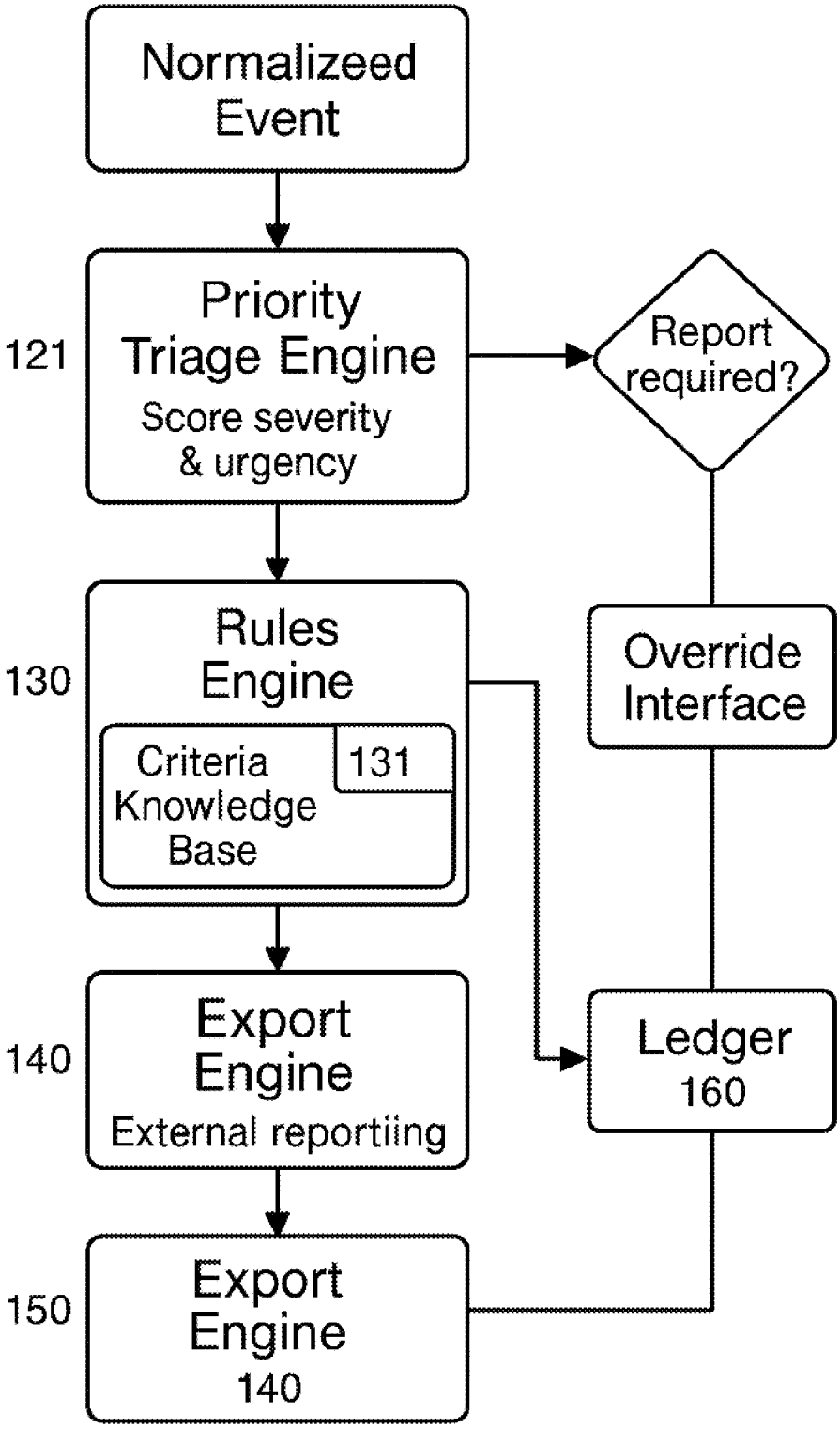

FIG. 2 – Decision Pipeline Flow

Figure 2: Decision Pipeline Flow, illustrating event triage, rule-based decision-making, and feedback loop. A **Priority Triage Engine (121)** first scores each normalized event by severity and urgency, assigning a risk priority level (e.g. flagging critical events for immediate response). Next, a **Rules Engine (130)** with a built-in criteria knowledge base **(131)**checks regulatory thresholds and PSQIA guidelines. It automatically decides which actions are required – e.g. whether the event must be reported to state health authorities, sent to a Patient Safety Organization (PSO) for confidential analysis, or both. After rules evaluation, the pipeline triggers two outputs in parallel: the event data is sent to the Export Engine for external reporting, and a log entry is added to the immutable Ledger (150) for audit. An **optional Clinician Override workflow (160)** is shown as a feedback loop: before final reporting, authorized staff may review or modify the system's decision via an **override interface (161)**. Any such human intervention is itself logged to the ledger, preserving a full audit trail of manual overrides. (Fig. 2 thus highlights decision points like compliance checks and override opportunities, as well as parallel logging of each step.)