JALUD EMBEDDED

SED Sapient Driver

BSI Flex 335 v2.0

V 4.1

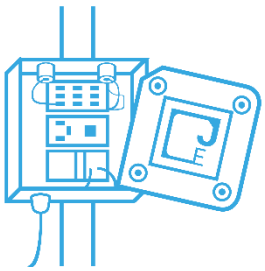

Table of Contents

Introduction

The Sound Event Detector (SED) integrates with external fusion and command-and-control systems utilizing the **BSI Flex 335 v2.0 / NATO STANREC 4869 v1.0 SAPIENT** protocol standards. Functionally, SED operates as a SAPIENT Autonomous Sensor Module (ASM).

The SAPIENT driver dynamically translates SED internal events such as hardware registrations, device statuses, and ambient sound event detections into continuous Protocol Buffers (Protobuf) streams for seamless multi-sensor fusion.

Low-Level Transport Architecture

SAPIENT messages are transmitted via a generic TCP socket connection. To overcome standard TCP stream defragmentation, the SED driver wraps every protocol message recursively using an explicit payload size indicator prepended directly into the binary array.

- **Packet Header:** A 4-byte unsigned integer specifying the exact byte size of the associated active Protobuf package.
- **Endianness:** The 4-byte header is encoded in Little-Endian format.
- **State & Framing:** Inbound data is accumulated and robustly framed using the 4-byte prefix. Sockets are safely decoupled on disconnects and seamlessly re-established upon device reconnection.

Identification & Identity Management

SAPIENT mandates strict adherence to ULID and UUID formats. SED utilizes deterministic, stateless hash generators to ensure consistent cross-session identification.

Node Entity (UUIDv5)

SAPIENT requires that the `node_id` matches an RFC 4122 compliant UUID. SED calculates a purely deterministic UUIDv5 hash upon initialization: * **Fixed Namespace:** SED utilizes a privately-declared generic SED-SAPIENT global namespace. * **Entity Factor:** Device serial numbers or hardware configurations act as the canonical string variable, ensuring zero-collision, persistent node IDs that survive network disconnections seamlessly without the necessity of database registries.

Reporting Artifacts (Base32 ULID)

Every Status cycle and outgoing Sound Detection transmits unique identity hashes targeting the `report_id` and `object_id` declarations. SED constructs a 26-character Crockford Base32 ULID out of a 48-bit epoch and 80-bit randomness, ensuring chronologically-sortable logging events on centralized SAPIENT fusion servers.

Logical Object Mapping

By design, SAPIENT targets traditional RF, Visual, or Kinetic implementations. SED translates Acoustic paradigms accurately to fit within the SAPIENT framework:

Registration Assertions (`Registration`)

On first connection, the ASM publishes its capabilities and hardware layout. * **Type Specification:** Registers accurately invoking `NODE_TYPE_ACOUSTIC`. * **Capabilities Framework:** Identifies broadly under the category `"Sensor"` alongside the specific type `"Audio Classification"`. * **Mode Operations:** Registers to run unconditionally bounded inside the `MODE_TYPE_DEFAULT` wrapper. The global surveillance area is registered as a generic `REGION_TYPE_AREA_OF_INTEREST`.

System Metrics (`StatusReport`)

- Sent systematically according to the configured heartbeat interval window.
- Reports active statuses under the flag `INFO_UNCHANGED` to prevent log bloat.
- Flags `SYSTEM_OK` natively for connected, non-faulty microcontrollers.
- **Dynamic Geolocation Updates:** Echoes system coordinates via `Location` structures (`LOCATION_COORDINATE_SYSTEM_LAT_LNG_DEG_M` and `LOCATION_DATUM_WGS84_E`). If the physical device shifts location via its GPS sensor, SED actively bypasses the standard heartbeat interval to instantly emit a new `StatusReport` signaling the position change.

Ambient Classifications (`DetectionReport`)

Detections resolve sound signals mapping strictly to SAPIENT expectations: * **Classifications:** The SED AI pipeline maps the neural confidence (%) directly to the `confidence` wrapper and feeds the active alarm category (e.g., Gunshot, Drone) directly into the `type`. * **Object Evidence:** URLs mapping directly to the audio blobs (.wav) and timeline spectrogram artifacts execute seamlessly via `AssociatedFile` structures in the payload.

ASM Tasking (`Task` -> `Command`)

The ASM asynchronously parses incoming SAPIENT instructions off the TCP bus. * **Registration Replay:** If the fusion node requests a re-registration cycle by invoking the

`Task` entity coupled with `command.request == "Registration"`, the driver evaluates the destination and instantly echoes the Registration handshake back to the network.

SAPIENT Taxonomy Mapping

SED rigorously translates native audio alarm labels into compliant SAPIENT taxonomic object types and sub-classes.

SED Native Class	SAPIENT `type`	Sub-Class (Level 1)	Sub-Class (Level 2)
Drone	Air vehicle	UAV rotary wing	-
Gunshot (all variants)	Equipment	Weapon	Gun
Scream (Female)	Human	Female	-
Male Scream, Aggression	Human	Male	-
People, Children	Human	-	-
Bird	Animal	Bird	-
Animal	Animal	Other	-
Vehicle, Traffic	Land vehicle	Other	-
Grinder	Equipment	Other	-
Other (Glassbreak, Siren)	Other	-	-

Protocol Message Examples (JSON)

The following examples demonstrate the serialized SAPIENT payloads translated into readable JSON format as implemented by SED.

DeviceRegistered (`Registration`)

Sent when a device first connects.


```
{
  "timestamp": "2026-04-28T12:00:00.000Z",
  "node_id": "8f075d9c-dc22-4467-8ddb-6bbfa948332d",
  "registration": {
    "name": "SED_ABC123",
    "capabilities": [
      {
        "category": "Sensor",
        "type": "Audio Classification"
      }
    ],
    "status_definition": {
      "status_interval": {
        "value": 30000.0,
        "units": "TIME_UNITS_MILLISECONDS"
      }
    },
    "mode_definition": [
      {
        "mode_name": "Default",
        "mode_type": "MODE_TYPE_DEFAULT",
        "task": {
          "region_definition": {
            "region_type": ["REGION_TYPE_AREA_OF_INTEREST"]
          }
        }
      }
    ]
  }
}
```


Heartbeat (`StatusReport - Online`)

Sent systematically based on the heartbeat interval, or instantaneously upon coordinate shifts.


```
{
  "timestamp": "2026-04-28T12:00:30.000Z",
  "node_id": "8f075d9c-dc22-4467-8ddb-6bbfa948332d",
```

```
  "status_report": {
    "report_id": "01HGWJXK9A...",
    "info": "INFO_NEW",
    "mode": "Default",
    "system": "SYSTEM_OK",
    "node_location": {
      "x": 14.42076,
      "y": 50.08804,
      "coordinate_system": "LOCATION_COORDINATE_SYSTEM_LAT_LNG_DEG_M",
      "datum": "LOCATION_DATUM_WGS84_E"
    }
  }
}
```

EventClassified (`DetectionReport`)

Sent immediately when an ambient acoustic alarm triggers. Example illustrates a *Gunshot* event mapping.

```
{
  "timestamp": "2026-04-28T12:15:22.123Z",
  "node_id": "8f075d9c-dc22-4467-8ddb-6bbfa948332d",
  "detection_report": {
    "report_id": "01HGWJXK9C...",
    "object_id": "01HGWJXK9D...",
    "location": {
      "x": 14.42076,
      "y": 50.08804,
      "z": 0.0,
      "coordinate_system": "LOCATION_COORDINATE_SYSTEM_LAT_LNG_DEG_M",
      "datum": "LOCATION_DATUM_WGS84_E"
    },
    "classification": [
      {
        "type": "Equipment",
        "confidence": 98.5,
        "sub_class": [
          {
            "type": "Weapon",
            "level": 1,
            "sub_class": [
              {
                "type": "Gun",
                "level": 2
              }
            ]
          }
        ]
      }
    ],
    "associated_file": [
      {
```

```
        "url": "http://<Server_IP>/media/Gunshot_260428-121522.wav"
      }
    ]
  }
}
```