# REMOTE HEALTH MONITORING ARCHITECTURE

## Design and Analysis of a Telerehabilitation Support System

Falbo Andrea 887525 Pellegrini Damiano 886261 Sanvito Marco 886493 Tenderini Ruben 879290

# OVERVIEW

01 Project Outline

02 Assumptions

03 Architectural Analysis

04 Problem Architecture

05 Logical Architecture

06 Concrete Architecture

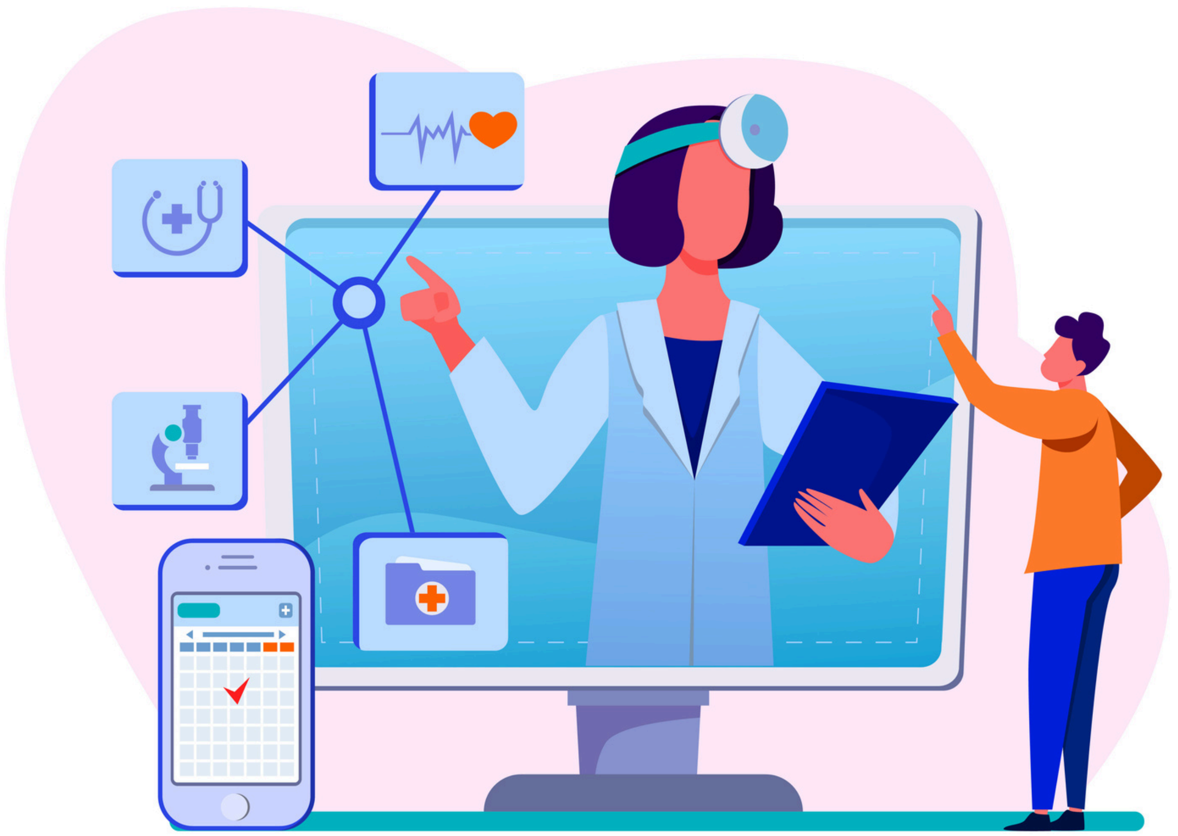
07 Deployment Architecture

08 Solution Evaluation

# Project Outline

# 01

# Objective



Design and implement a system for remote health monitoring and telerehabilitation for patients preparing for surgical interventions.

The system will track physiological parameters and daily activities, including adherence to rehabilitation plans. Alerts are generated if thresholds are exceeded, prompting the attending physician to contact the patient.

01

# System Requirements 1

Real-time Data Acquisition: Collect sensor data in real-time according to the patient's therapeutic plan.

Therapeutic Plan Management: Support physicians in updating the treatment plan, including adjusting the frequency of physiological parameter monitoring.

Automatically implement any changes to the therapeutic plan.

Anomaly Detection: Monitor physiological parameters to identify abnormal values outside predefined thresholds.

In case of anomalies, identify an on-duty physician for an in-person visit.

01

# System Requirements 2

For critical alarms (e.g., “red code”), locate the nearest ambulance and hospital for patient transport.

Physician Notification: Notify the assigned physician and provide the patient’s medical record.

Data Reporting: Allow the on-duty physician to send recorded parameters, diagnosis, and other health-related information.

Activity Tracking: Acquire information from the tracking platform regarding daily activities performed by the patient.

Verify at the end of the day whether rehabilitation exercises were completed as prescribed.

01

# Assumptions

02

# Operational Context

The system operates within the Lombardy region and integrates with local healthcare infrastructure.

Compliance with Italian and European regulations, including GDPR, medical device certification, and telemedicine guidelines.

Integration with Lombardy's healthcare network: 8 ATS, 27 ASST, ~180 hospitals, and ~1,200 specialist clinics. [1]

Geographic patient distribution: 35% Milan, 25% Bergamo-Brescia, 15% Como-Varese, 25% other provinces.
Target population: high-risk pre-surgical patients (~50,000–70,000 annually).

02

# Technology Integration

02

Support for heterogeneous physiological monitoring devices, without managing device lifecycle. [2]

Two sensor types: automatic (continuous) [3] and manual (patient-input) sensors.

Integration with third-party activity tracking platforms for daily activity summaries and alerts. [4]

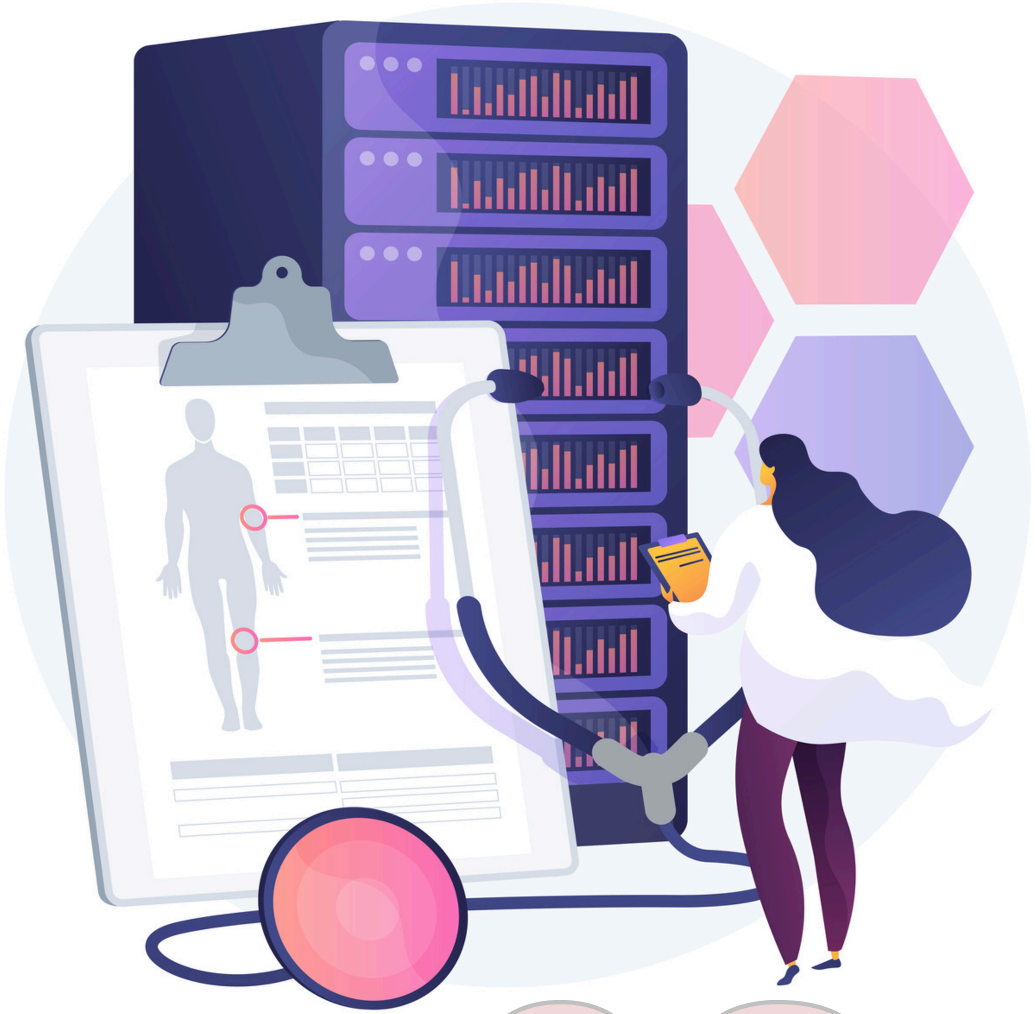
# Frequencies

Physiological Monitoring per patient:

- Automatic sensors: ~178 values/day
  - Heart rate: every 15 minutes
  - Blood pressure: scheduled measurements
  - Oxygen saturation: every 30 minutes
  - Temperature: every 2 hours
  - ECG: anomalous events and daily summary
- Manual parameters: ~15 values/day
- Data volume: 20-65 KB/patient/day

Activity Tracking:

- Daily metrics: 18 values/day
- Data volume: 2-5 KB/patient/day



02

# Performance

02

System-wide Processing Requirements

- Total data throughput: 571,500 values/hour
- Peak periods: 15-20% above average during morning and evening
- Emergency scenarios: 350-700 red codes daily
- Anomaly: 6.000 daily
- Threshold exceeded: 16.000 daily

The system must comply with healthcare data protection, medical device regulations, and telemedicine standards, without compromising emergency response performance.

# Architectural Analysis

# 03

# Significant Requirements

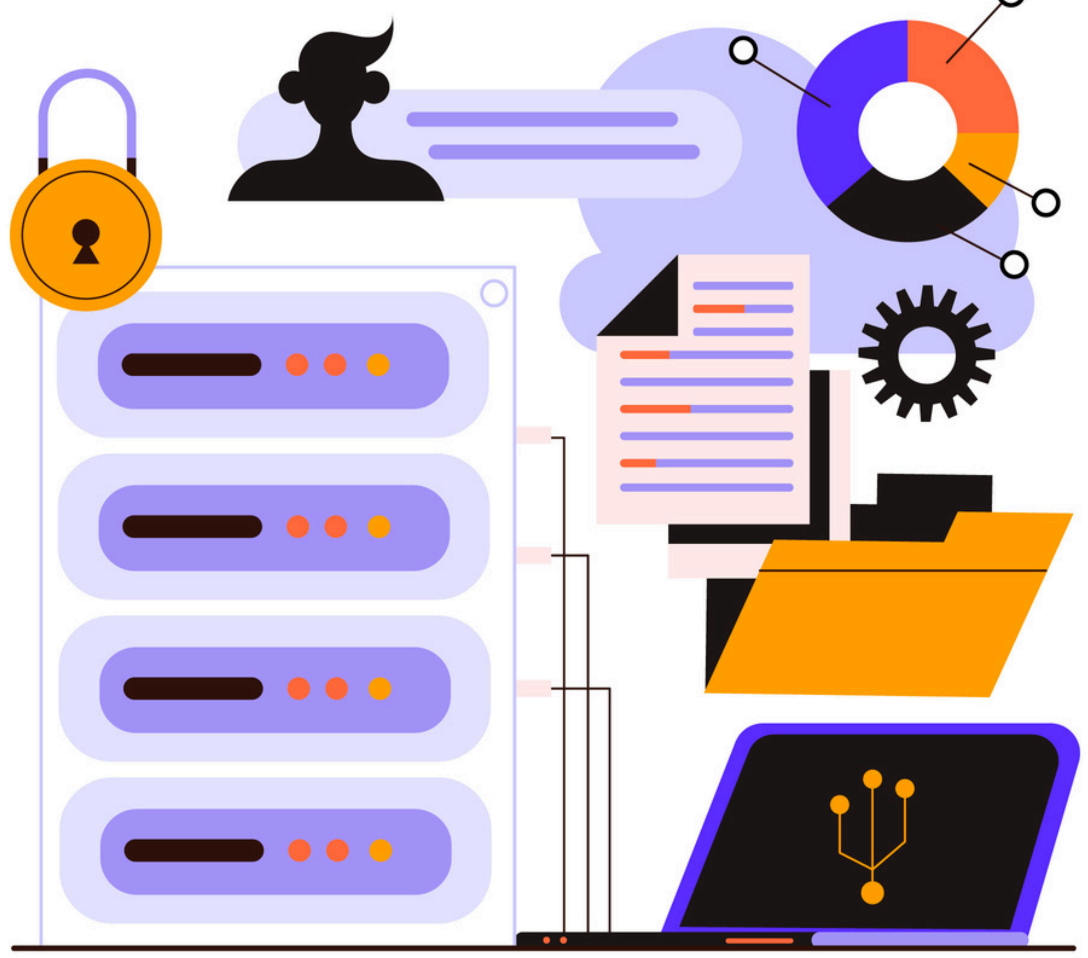
Unified data processing priority-based pipeline:

- Red code emergencies: < 1 min detect, < 2 min notify
- Standard anomalies: < 5 min detect, < 2 min notify
- Threshold violations: < 30 min detect, < 2 min notify

Activity platform alerts include emergency situations and behavioral anomalies.

The system must support runtime modification of monitoring frequencies and thresholds across distributed patient populations, without downtime.

The processing architecture must handle concentrated loads while maintaining consistent service across the region.



03

# Significant Requirements

03

Availability: 24/7 operation, supporting up to 70,000 concurrent patients across distributed healthcare infrastructure, with fault tolerance for life-critical scenarios.

Performance:

- Peak throughput: 13.72 million values/day, with geographic load concentration
- Emergency response: < 2 minute end-to-end latency for critical alerts
- Concurrent sessions: thousands of simultaneous monitoring sessions across regional facilities

Scalability: Horizontal scaling to accommodate varying loads, from individual monitoring to regional deployments, with capacity planning for emergency surges.

Interoperability: Seamless integration across heterogeneous ecosystems (sensor networks, activity platforms, healthcare information systems, emergency services) using industry standards.

Security and Reliability: Medical-grade data protection and fault-tolerant operation during mission-critical emergency responses.

# Key Architectural Components

Multi-source Data Acquisition Partition: real-time ingestion from dual data streams (physiological and activity)

Hierarchical Event Processing Engine: geographic load balancing reflecting regional distribution

Alert Orchestration System: emergency dispatch, attending physician and on-call physician notification

03

# Constraints and Trade-offs

03

Hybrid Deployment: Centralized coordination across hospital hubs with distributed edge nodes improves latency and resilience, but increases complexity in orchestration and monitoring.

Event-Driven Processing: Multi-source data ingestion favors event-driven patterns for concurrency and responsiveness, but requires careful handling of ordering, buffering, and fault tolerance.

Redundancy vs. Cost: Hot spare nodes and shared storage increase availability and fault tolerance, but come with higher infrastructure and operational costs.

Security vs. Usability: HTTPS and VPNs ensure secure communication and patient data protection, but may add overhead and reduce ease of access for physicians.

Scalability vs. Consistency: Distributing components across hospitals and edge nodes allows scaling with patient load, but introduces challenges in keeping data consistent across sites.

# Data Consideratons

Although not part of the system's core requirements, decentralized healthcare architectures demonstrate that distribution is not a limitation but an enabler. The EU-funded BETTER project shows how large-scale, multi-source health data can be securely analyzed across borders using AI-driven tools, all while remaining fully compliant with GDPR. [5]

Similarly, the European Health Data Space (EHDS) initiative establishes a common framework for citizens and professionals to securely access and exchange health data across the EU.

These examples highlight that distributed processing can effectively support both individual patient monitoring and population-level analytics, combining innovation with strict privacy and regulatory compliance. This strengthens the case for adopting a distributed approach in remote healthcare monitoring systems.

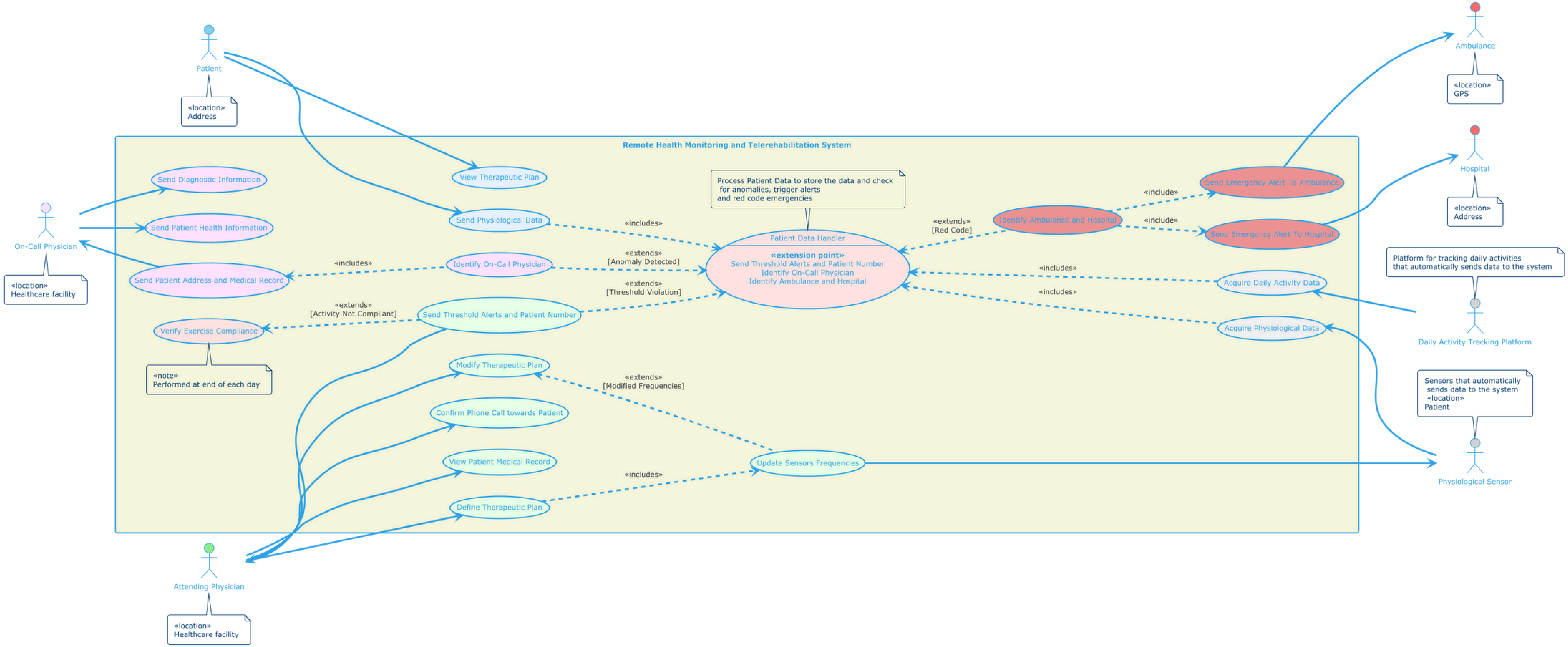
03

# Problem Architecture

04

# Use Case Diagram

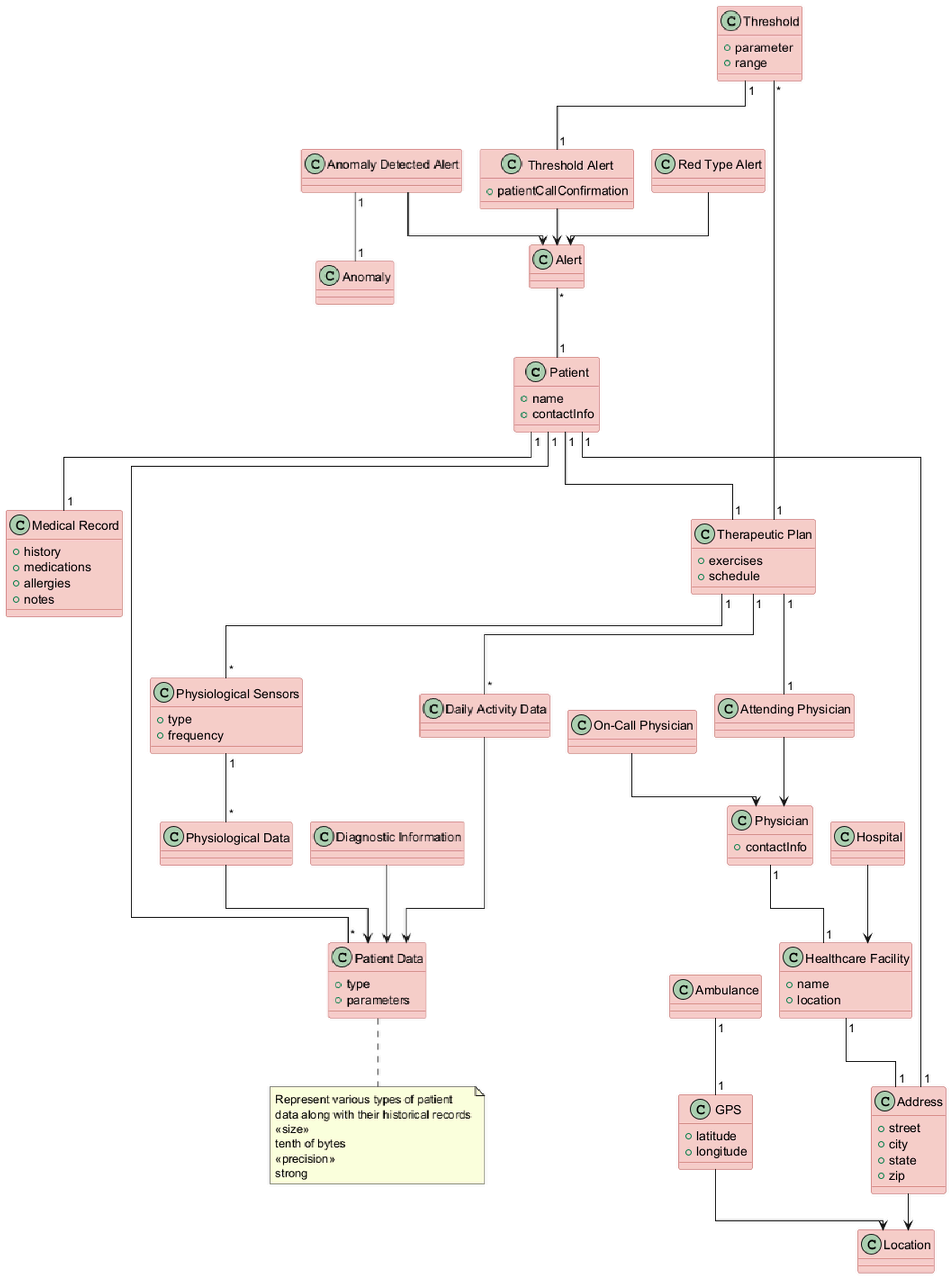
The use case diagram addresses the who and the where. The who is shown with actors and their use cases. The where is included for each actor with a note, except the Daily Activity Tracking Platform as it does not have a real location.

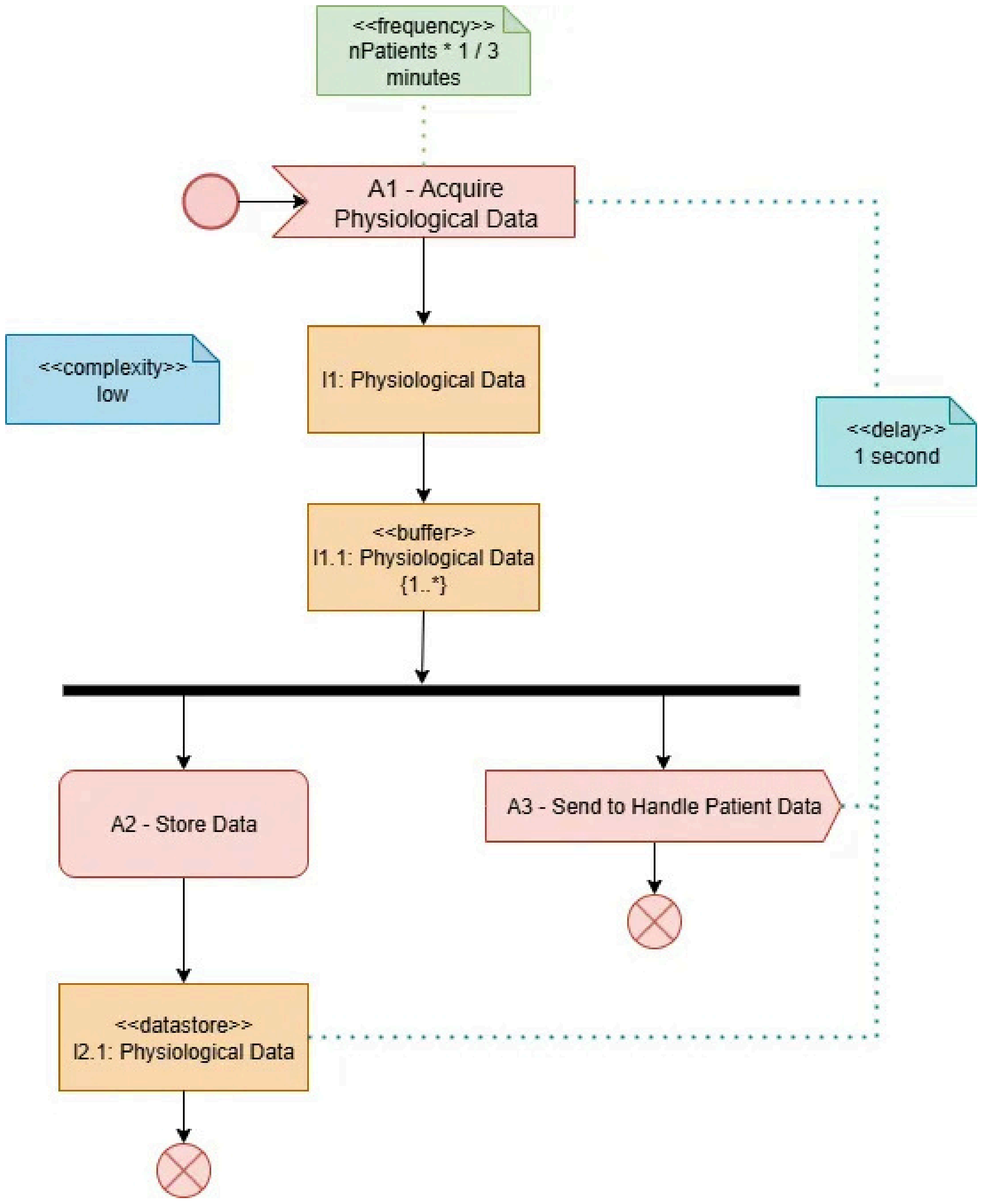


# Data Diagram

The data diagram for the problem architecture represents the what, placing the patient — the main actor of our system — at the center, together with their related information, including the therapeutic plan and different types of alerts.

The system also addresses the where through allocation: addresses and GPS coordinates represent the location of the actors within the system.



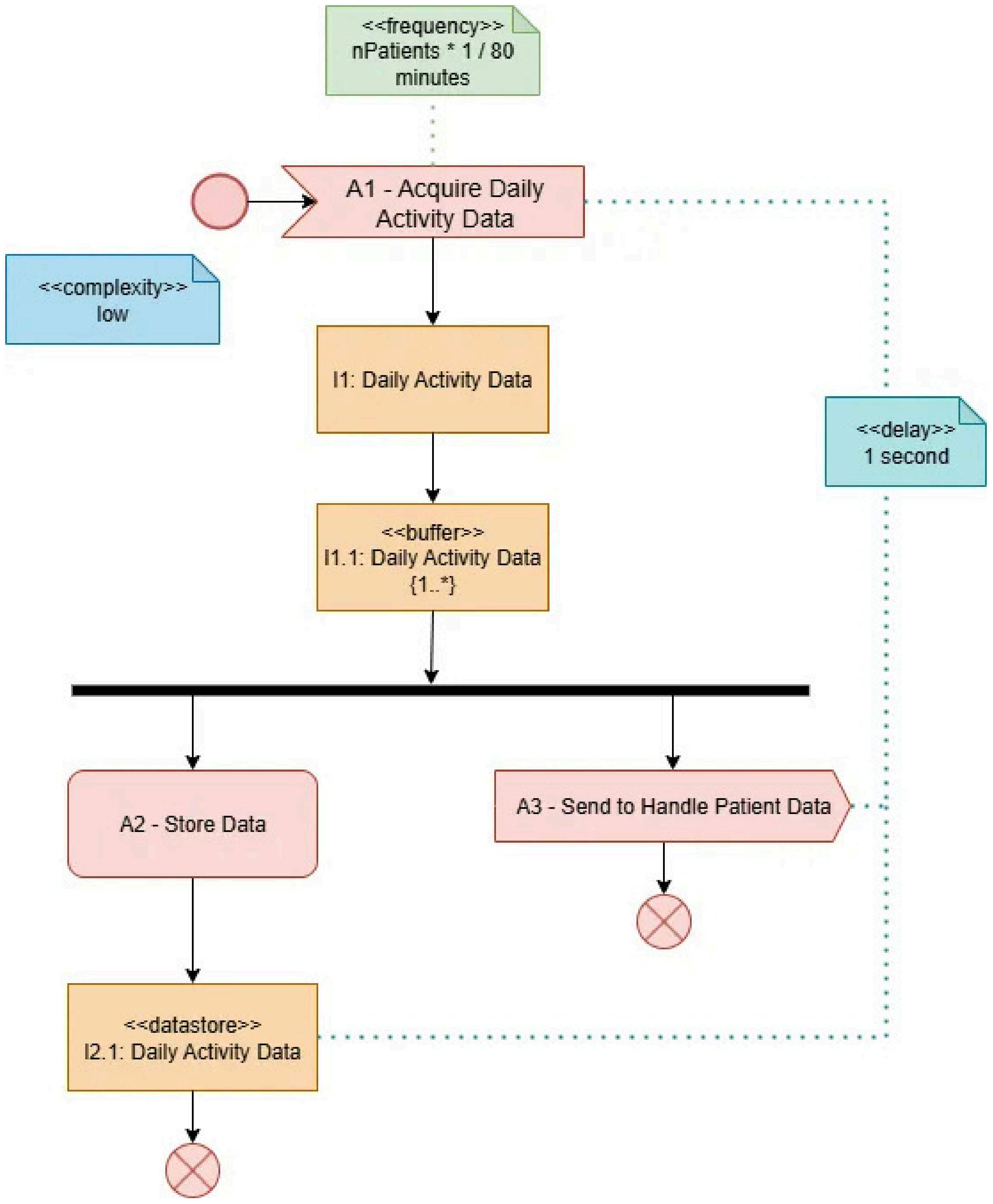



# ADUC 1: Acquire Physiological Data

The sensors send data to the system (not at fixed frequencies — we estimate about one reading every three minutes per patient, with high variability).

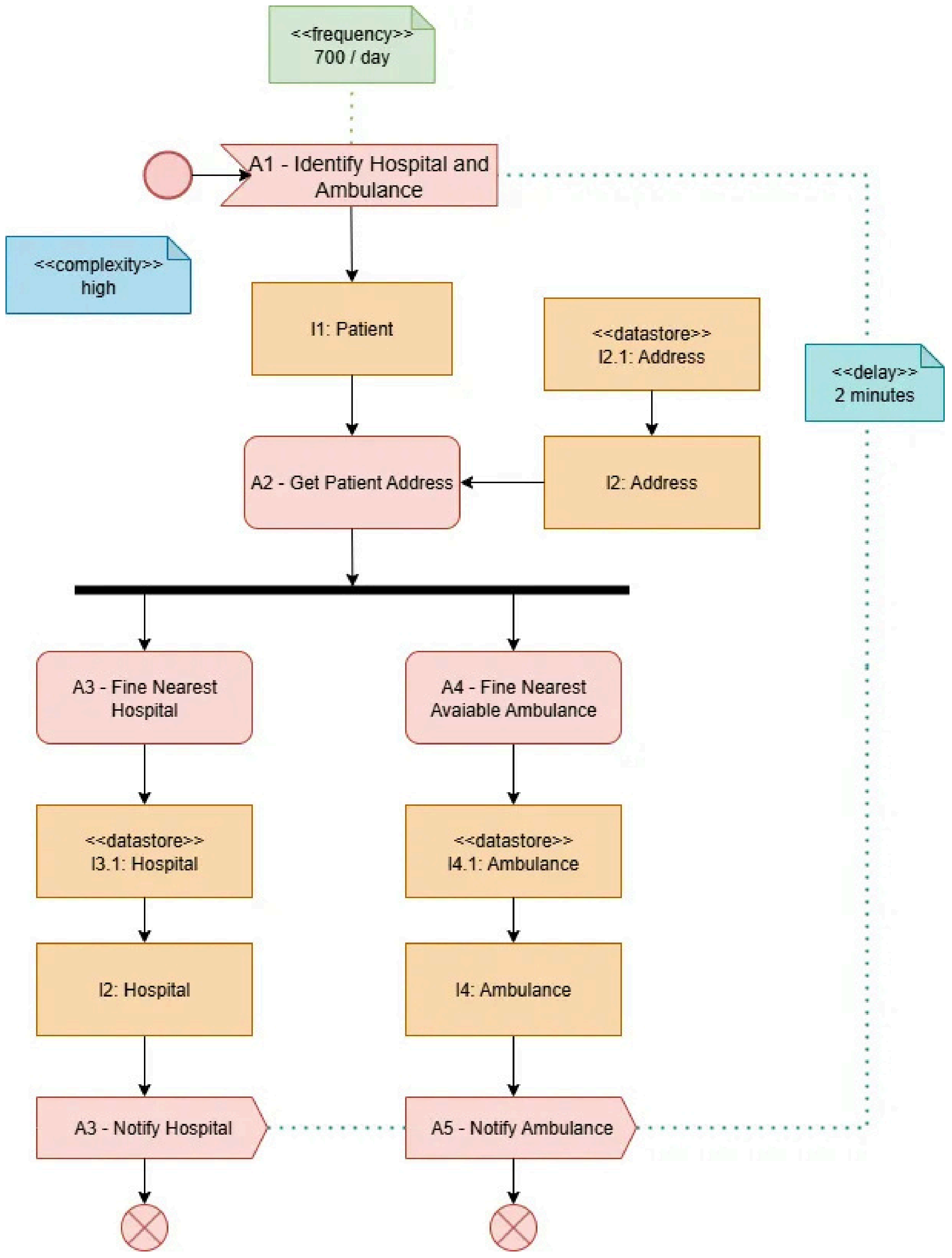
Each incoming data point is placed into a buffer to prevent data loss.

From there, the data is asynchronously stored, while in parallel an event manager dispatches a signal containing the data, which is then processed accordingly.




# ADUC 2: Acquire Daily Activity Data

Similarly to ADUC 1, the daily activity data — collected at a lower frequency (about once every 80 minutes per patient) — undergoes the same processing as the physiological data: buffered to avoid loss, asynchronously stored, and dispatched by the event manager.

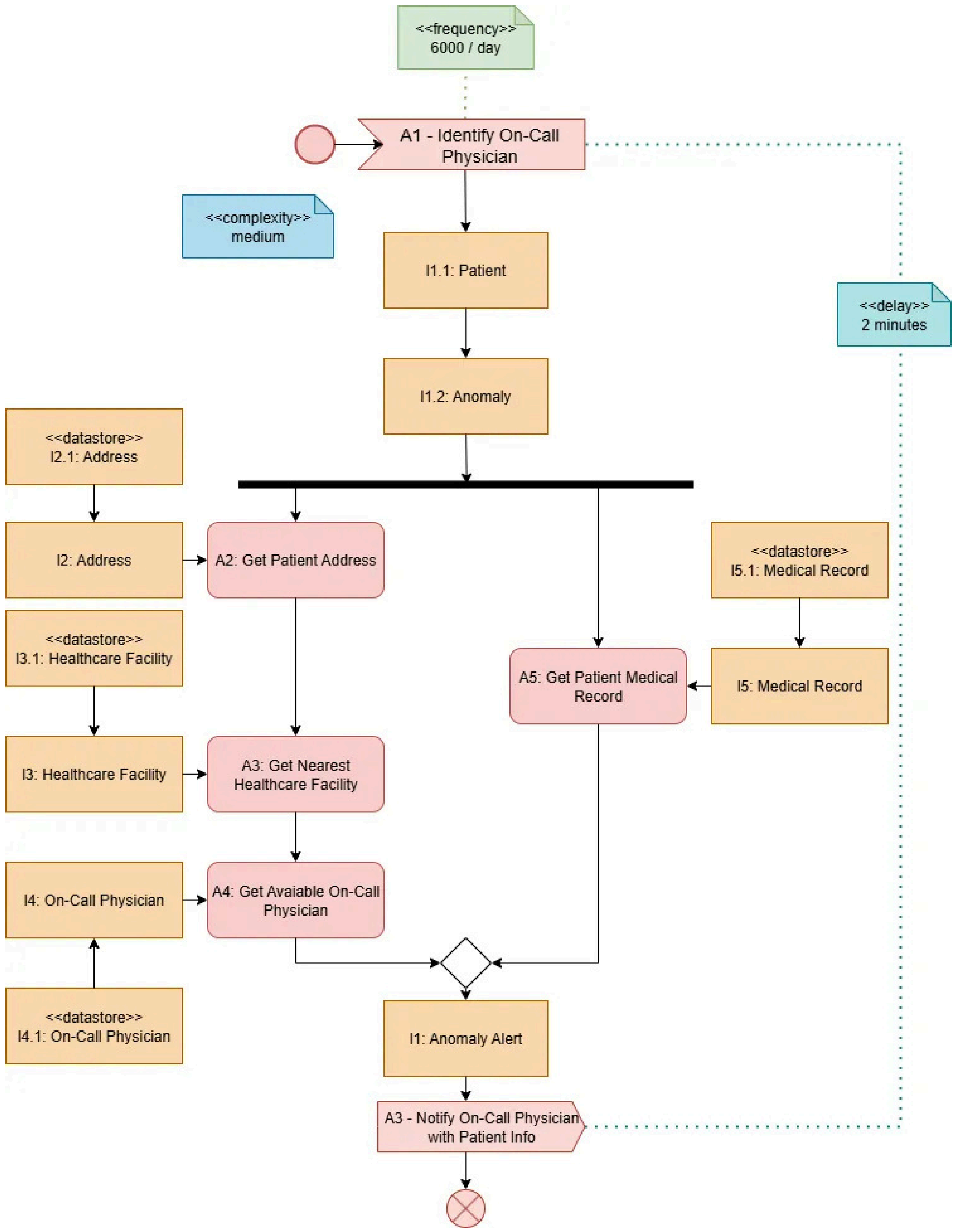



# ADUC 3: Identify Hospital and Ambulance

After receiving a red-code alert, the system retrieves the patient's address and, in parallel, identifies the nearest ambulance and hospital to notify them of the emergency.

The expected frequency of such events is relatively low compared to the total number of patients, approximately 700 cases per day, while the delay needs to be less then 2 minutes.

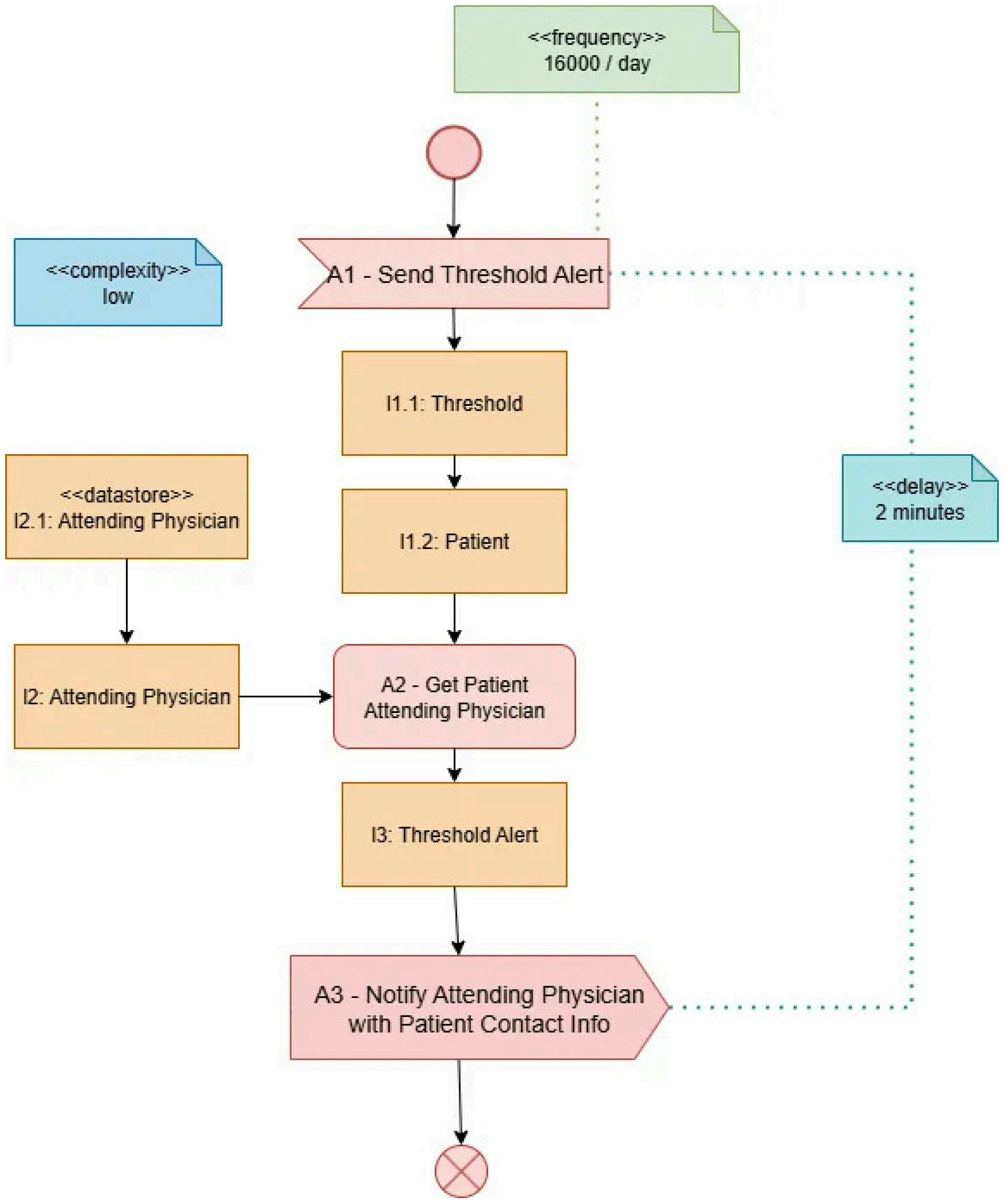
# ADUC 4: Identify Oncall and Notify




After detecting an anomaly, the system locates the nearest available on-call physician and delivers the patient's medical records along with the information needed to reach them.

We expect a 6000/day frequency and a 2 minutes delay.

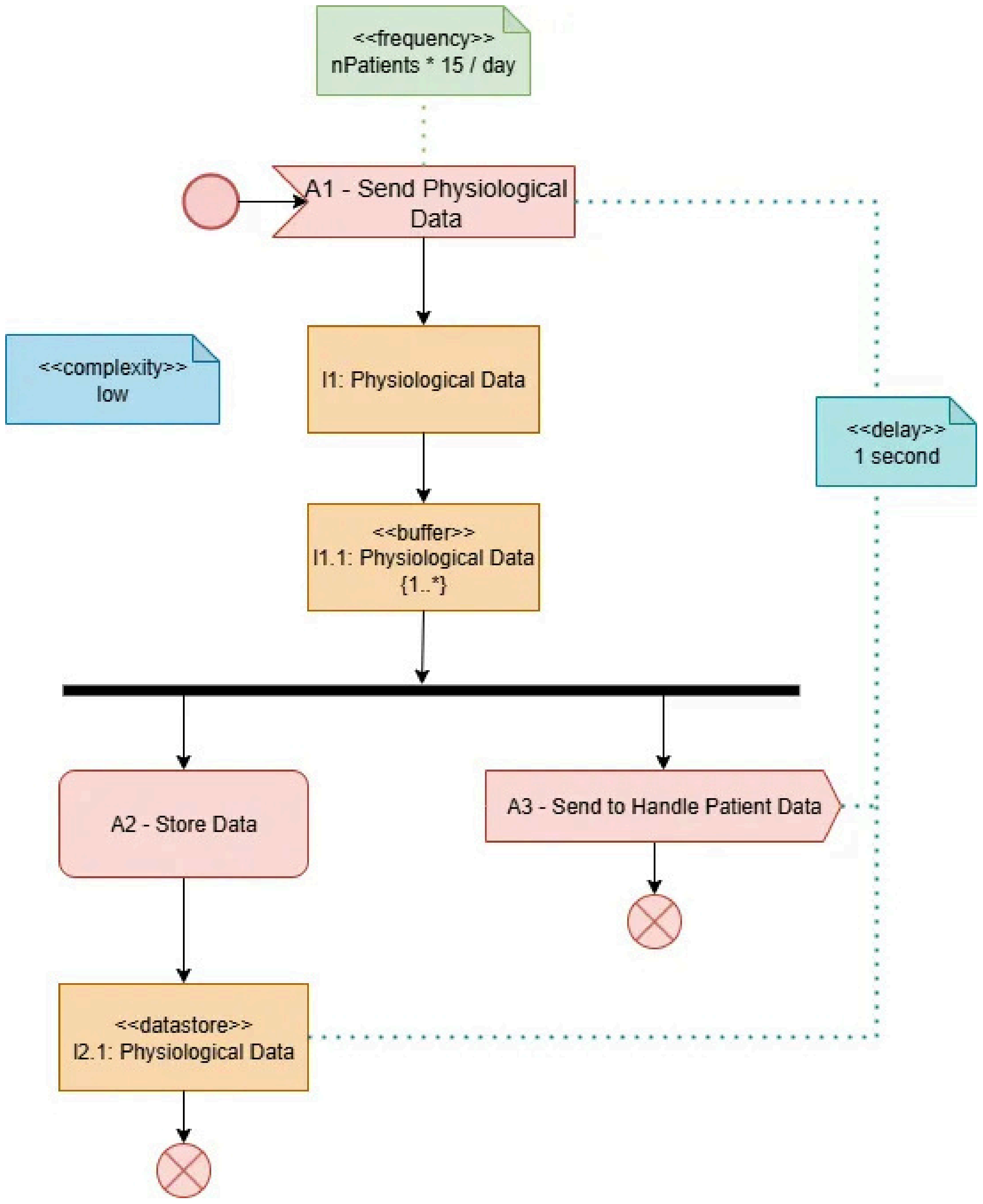
04




# ADUC 5: Send Threshold Alert

When a threshold is exceeded or not met, the system notifies the attending physician by sending the patient's contact information together with the alert signal.

This event is expected to occur about 16,000 times per day, with a delay of less than 2 minutes.



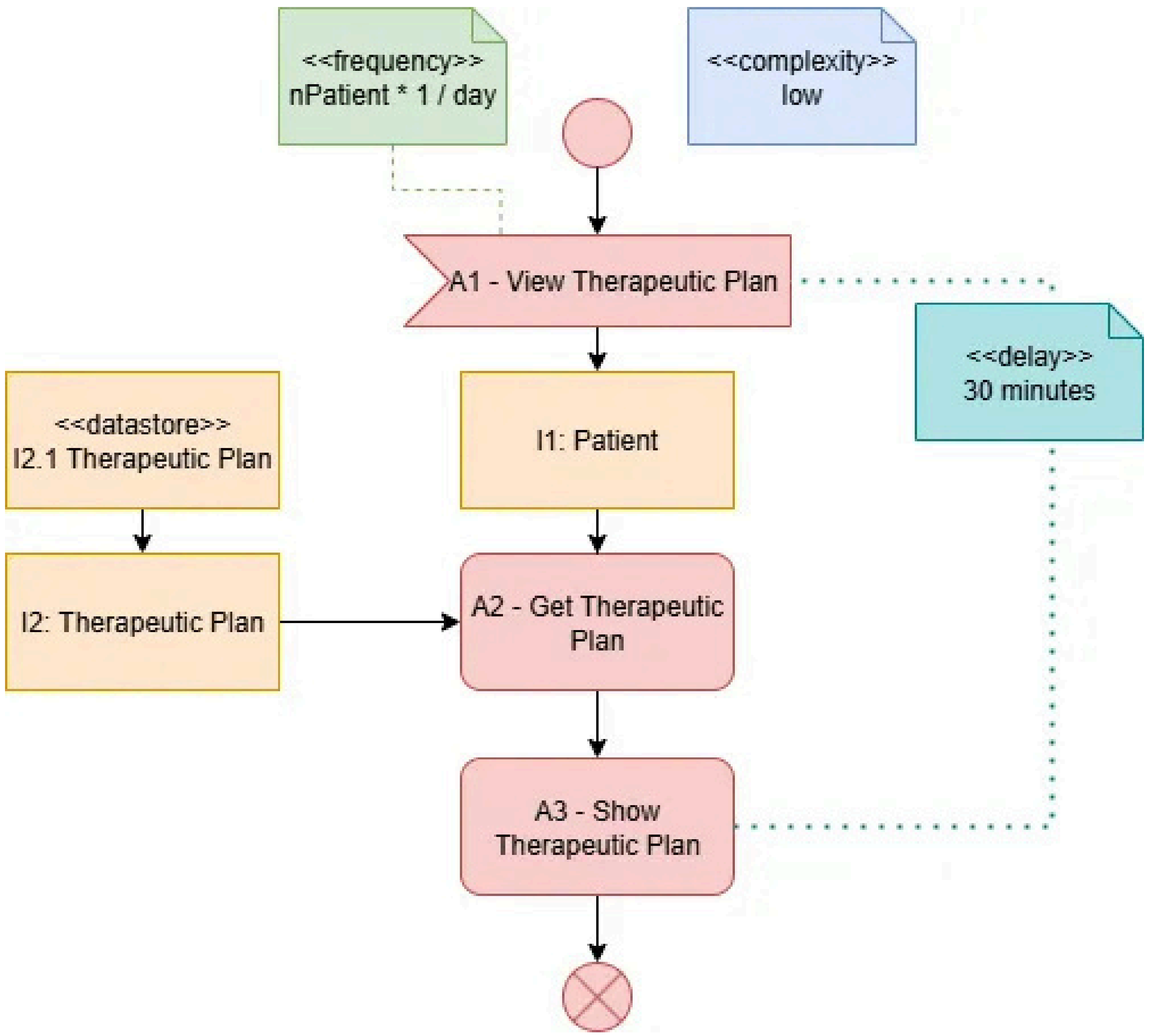

# ADUC 6: Send Physiological Data

Patients can also manually send physiological data.

Compared to sensor data, these inputs are much less frequent (about 15 per day per patient). T

hey are processed in the same way as sensor-based physiological data — buffered to prevent loss, asynchronously stored, and dispatched by the event manager.

04

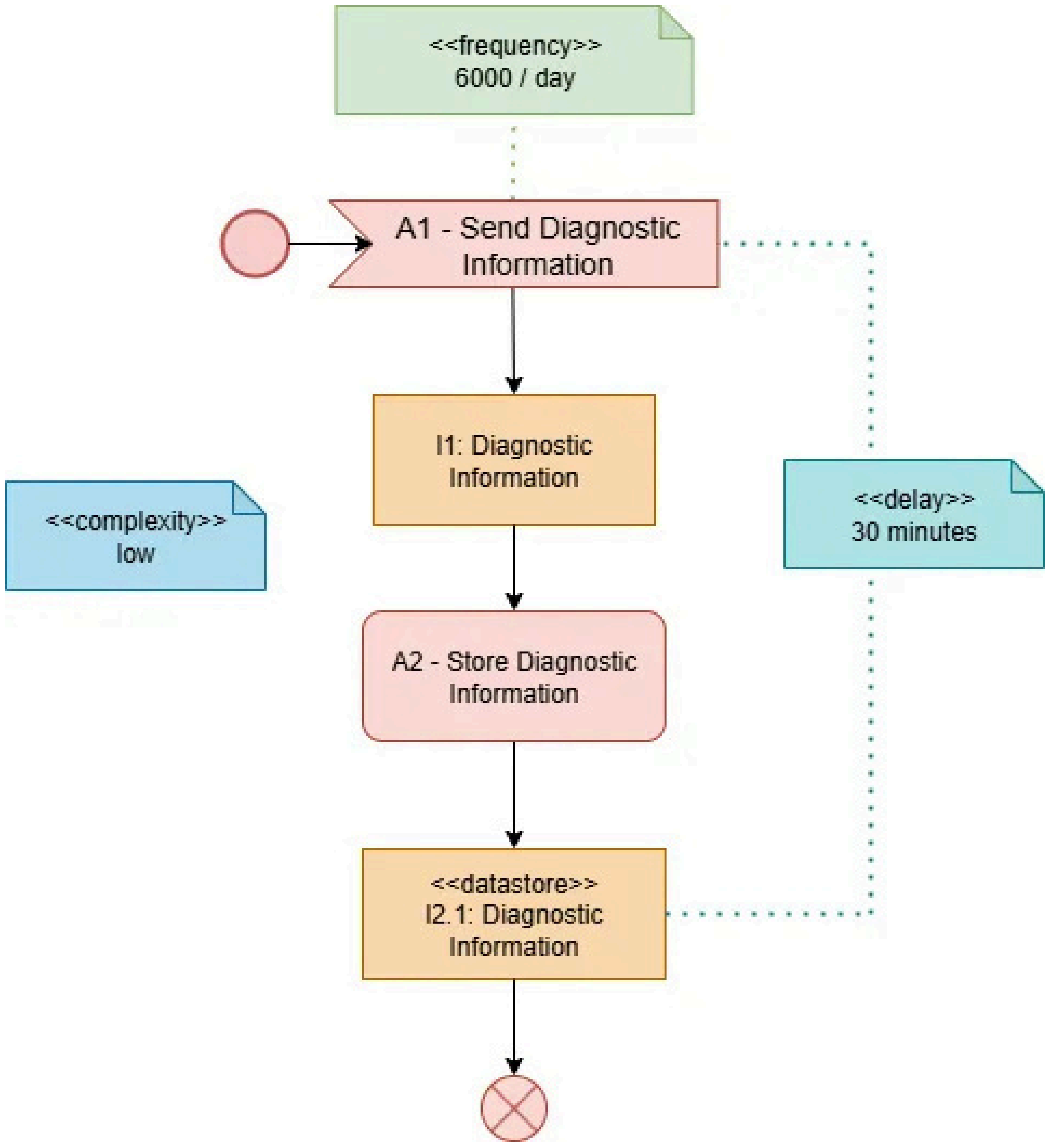



# ADUC 7: View Therapeutic Plan

The system allows patients to view their therapeutic plan.

This is a very rare event, expected roughly once per day per patient.

Given its infrequency, rapid response is not required, and a delay of up to 30 minutes is acceptable.

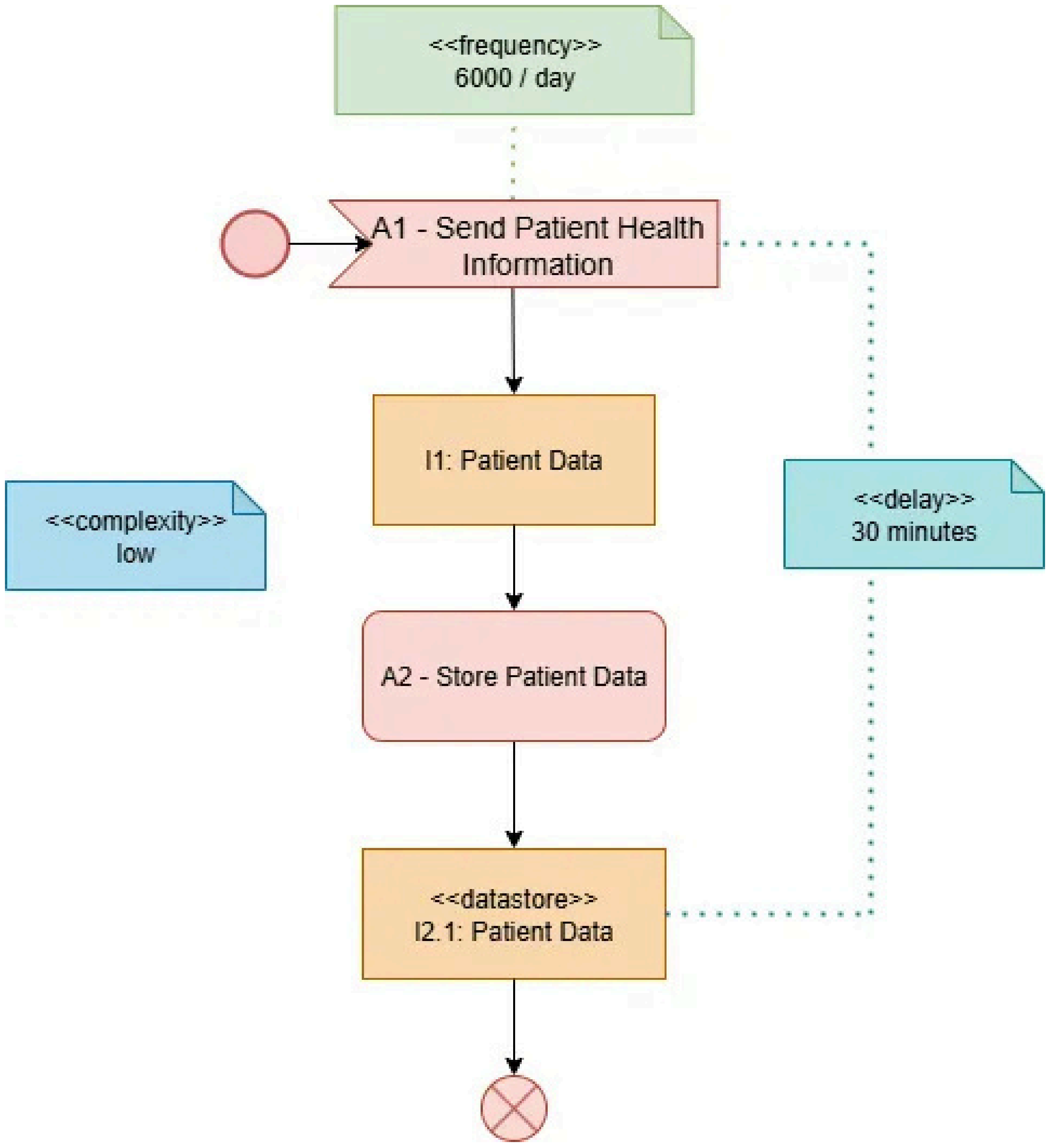



# ADUC 8: Send Diagnostic Information

After seeing the patient from an anomaly alert, the on-call physician must send diagnostic information as proof of the visit.

This activity is infrequent (around 600 times per day), so a high delay is acceptable.

04

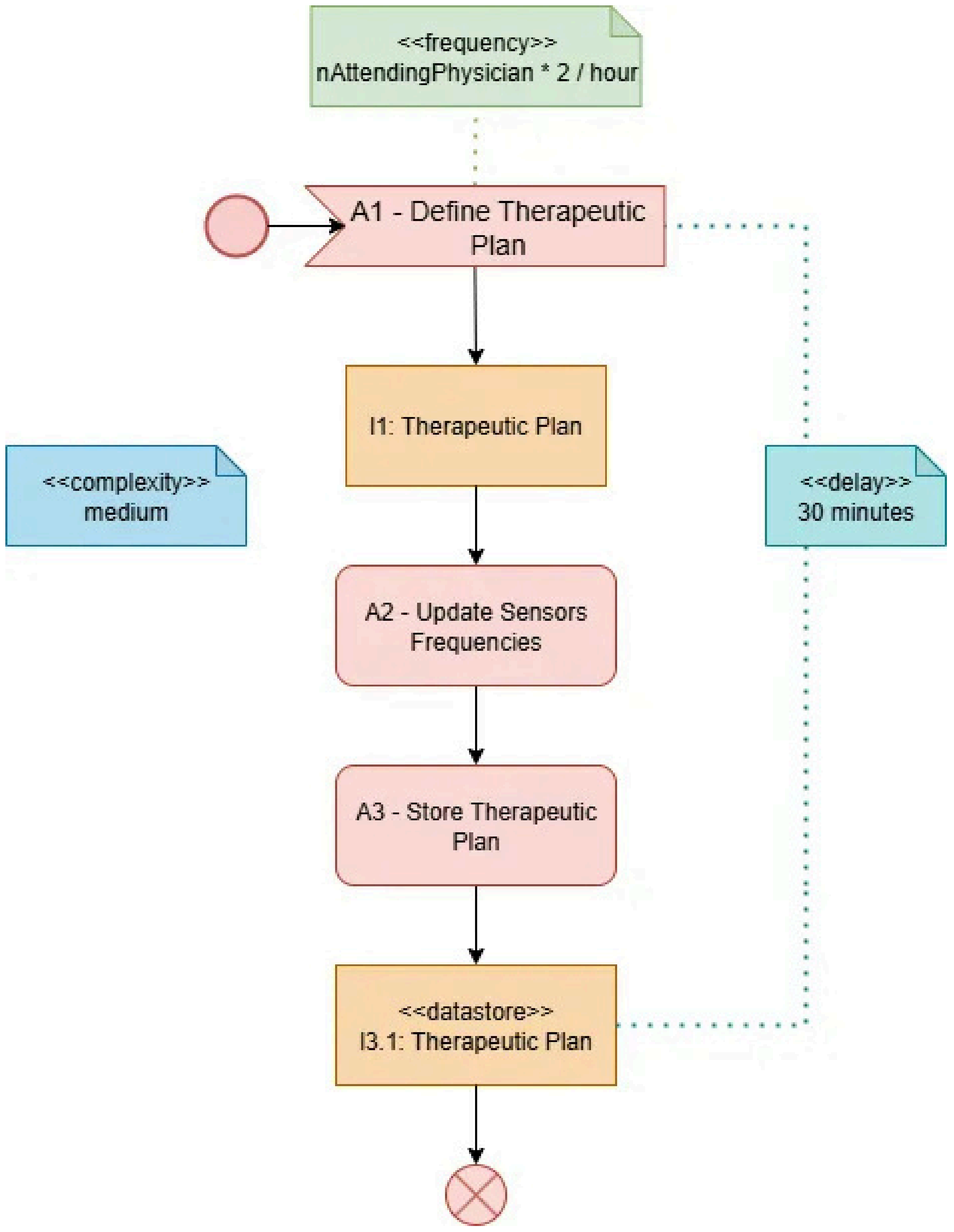



# ADUC 9: Send Patient Health Information

Similarly to ADUC 8, where diagnostic information is sent, here the on-call physician must also transmit patient health information.

The frequency and acceptable delay remain unchanged.

04

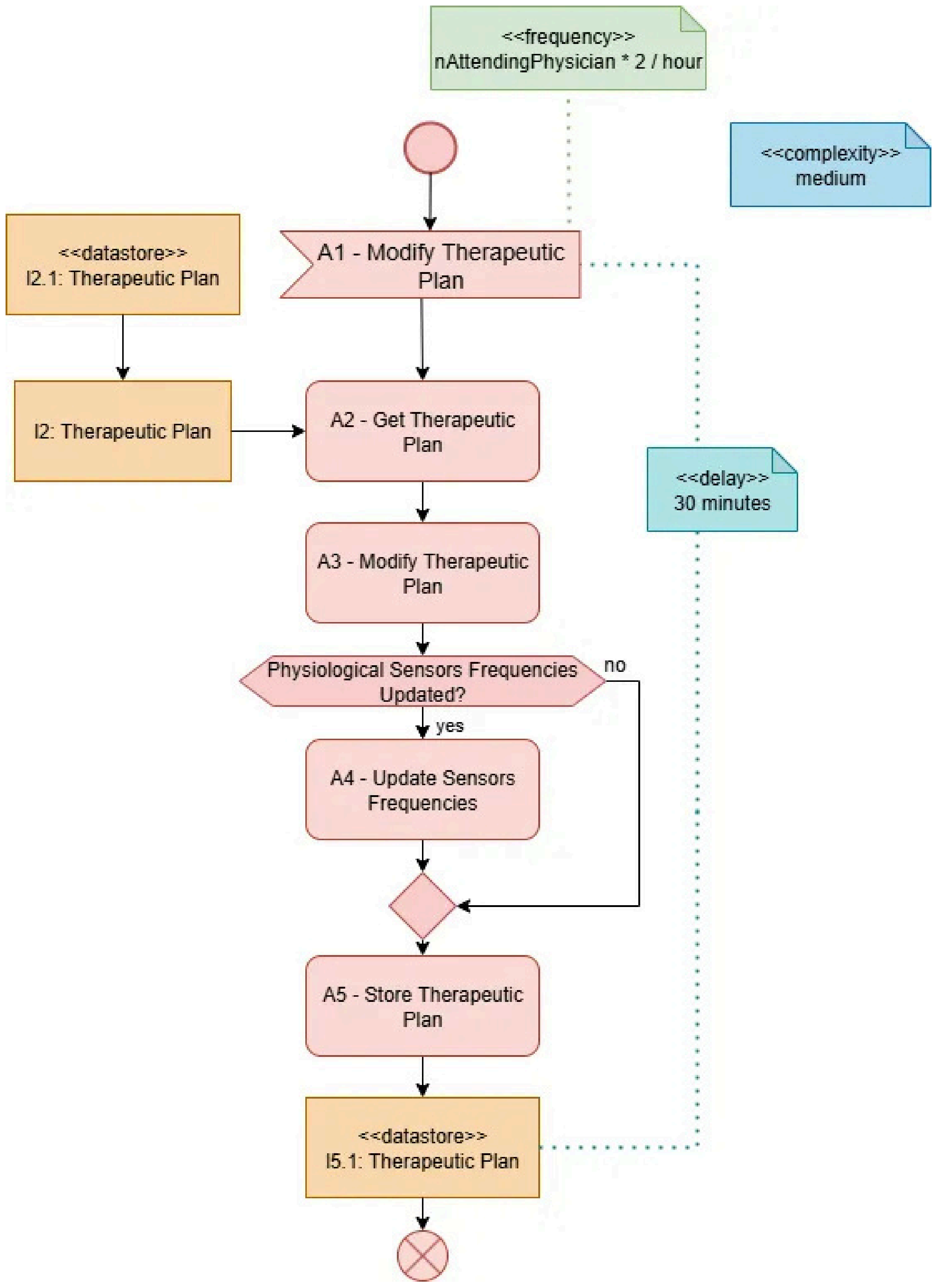



# ADUC 10: Define Therapeutic Plan (+ ADUC 13)

The system must allow the attending physician to define the therapeutic plan and, in doing so, configure the required sensor frequencies.

As these are low-frequency operations — up to 2 per hour per attending physician in the worst case — a high delay is acceptable.

04



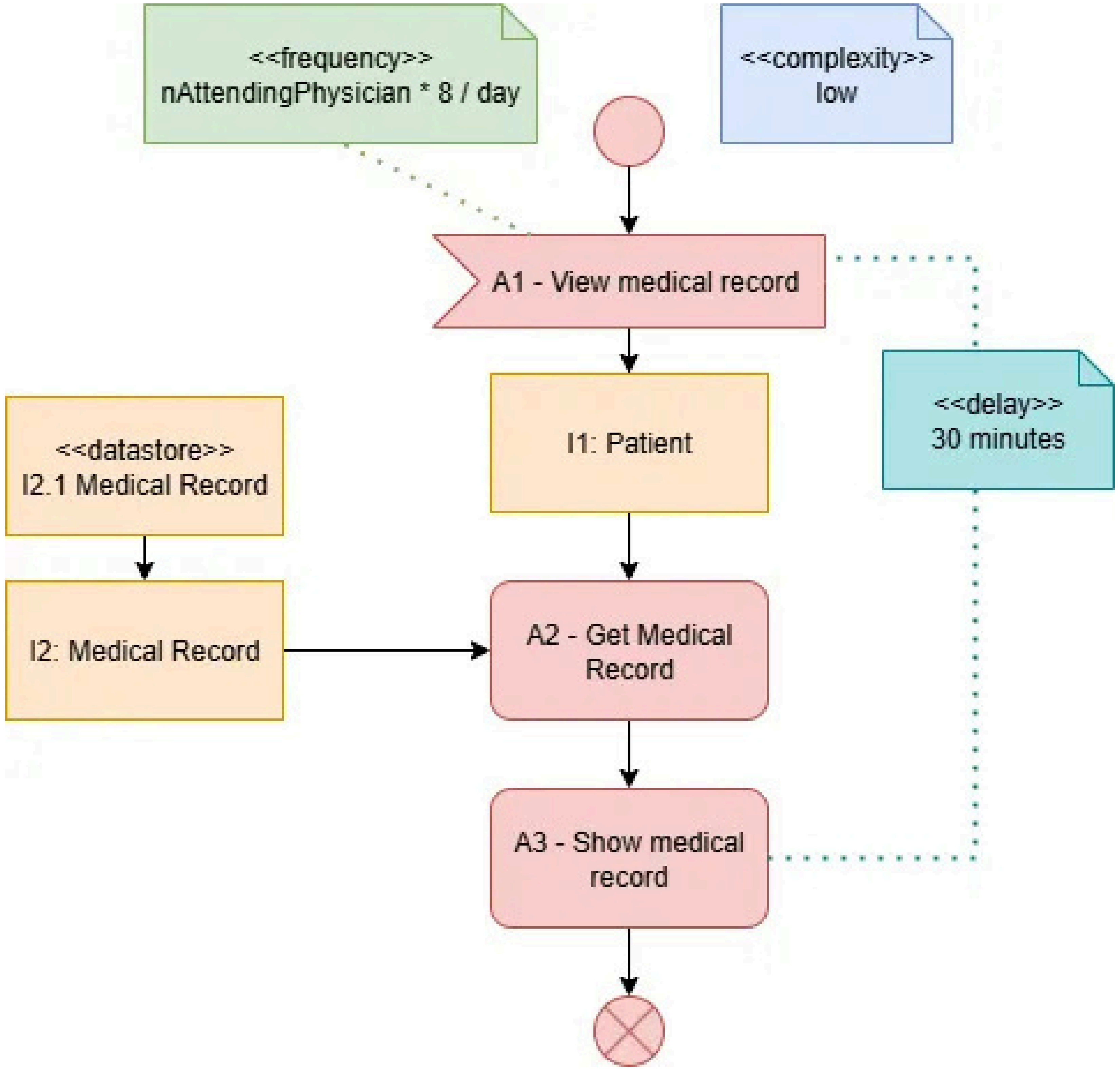

# ADUC 11: Modify Therapeutic Plan (+ ADUC 13)

Similarly to ADUC 10 which define the therapeutic plan, the system must allow the attending physician to modify the plan and update the sensor configurations accordingly.

Consequently, the frequencies and acceptable delays remain the same

04

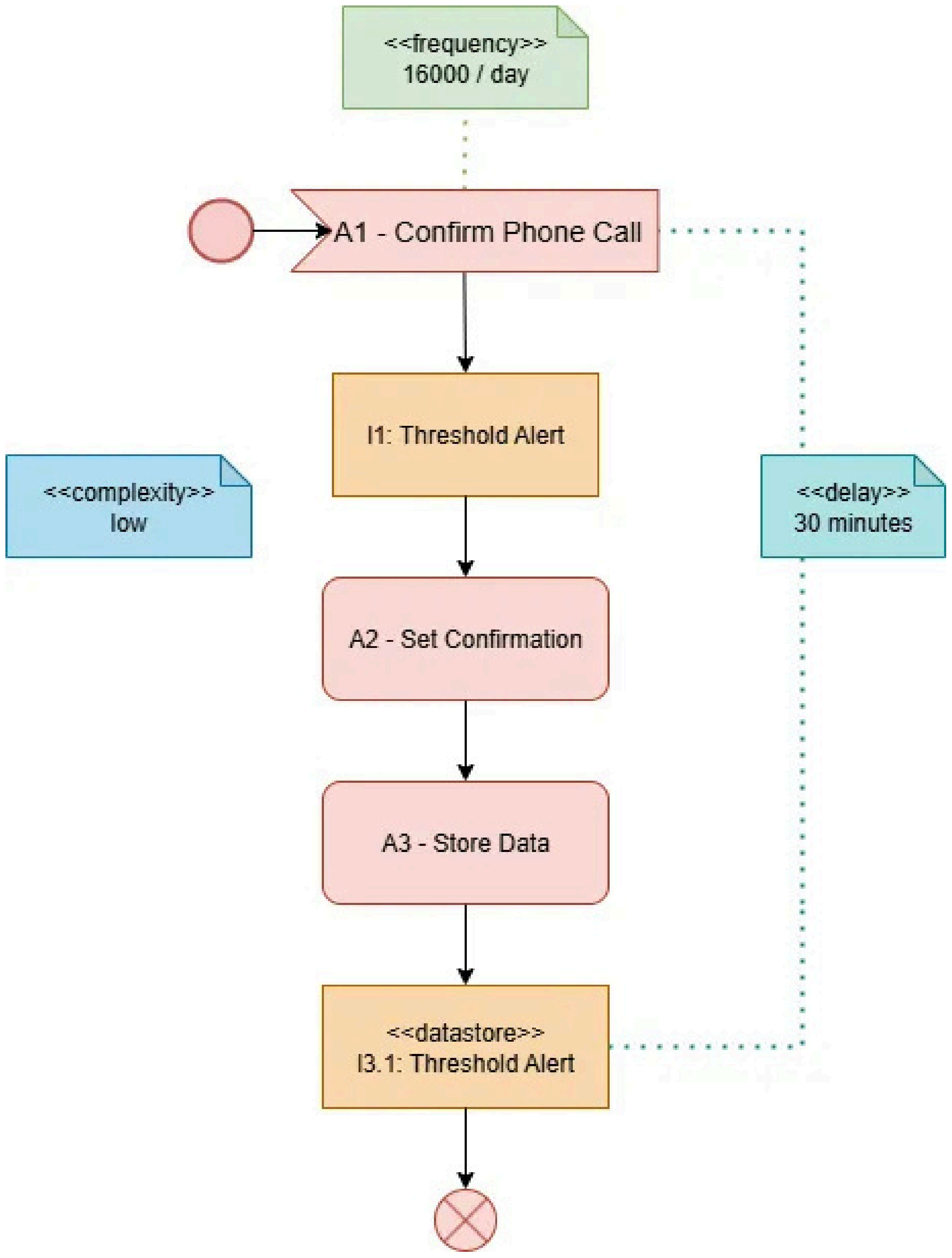
# ADUC 12: View Medical Records




The system must allow the attending physician to view the patient's medical records.

This is a low-frequency activity (about 8 times per day per physician), so a high delay is acceptable

04

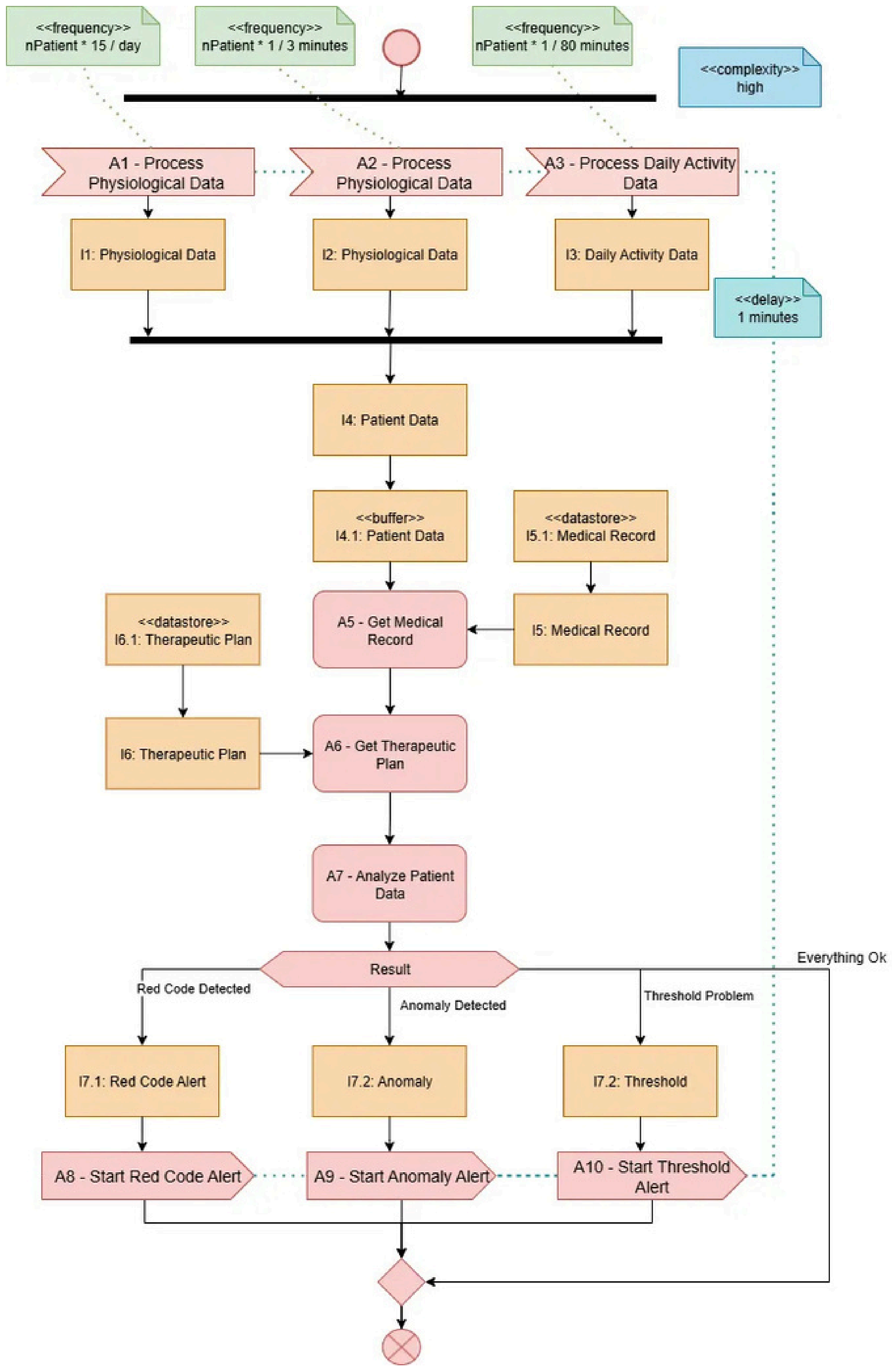



# ADUC 14: Confirm Phone Call

After a threshold violation alert, the attending physician must confirm the call to the patient for accountability purposes.

This activity is not very frequent relative to the number of patients, so a high delay is acceptable.

04

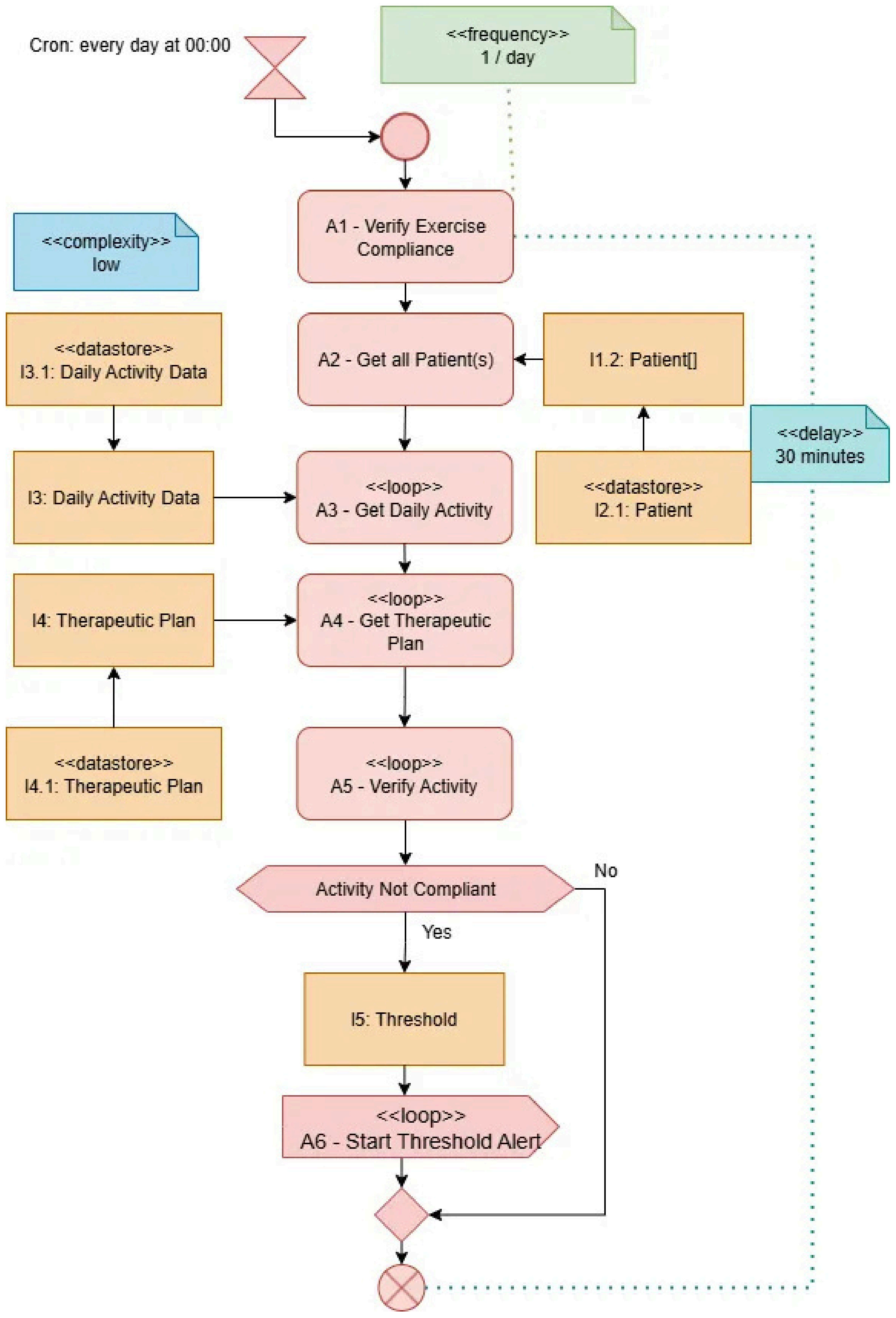



# ADUC 15: Patient Data Handler

This is the core of the system, where data is ingested from three sources: the two physiological streams (one patient-reported, one from sensors) and the daily activity data.

It analyzes incoming data to detect anomalies, red-code alerts, anomaly and threshold violations, triggering the corresponding alert events when necessary.

The system must continuously handle high-frequency data streams, so processing must be fast, with a maximum delay of 1 minute.




# ADUC 16: Verify Exercise Compliance

This activity runs daily at midnight and checks whether the patient has completed the activities assigned in their therapeutic plan.

If any activity is missed, a threshold alert is triggered to notify the attending physician.

Since this activity occurs once a day and it is not urgent, a high delay is acceptable.

# Logical Architecture

05

# Solution A. Activities

Functional partitioning divides the system into components, each responsible for a specific business function. This approach enhances cohesion, reduces coupling, and improves maintainability and scalability.

| Activity | Frequency | Complexity | Delay | Component |
|---|---|---|---|---|
| ADUC 1: Acquire Physiological Data | 1 / 3 minutes * #nPatient | low | < 1 second | C1 |
| ADUC 2: Acquire Daily Activity Data | 1 / 80 minutes * #nPatient | low | < 1 second | C1 |
| ADUC 3: Identify Hospital | 700 / day | high | < 2 minutes | C2 |
| ADUC 4: Identify Oncall and Notify | 6000 / day | medium | < 2 minutes | C2 |
| ADUC 5: Send Threshold Alert | 16000 / day | low | < 2 minutes | C2 |
| ADUC 6: Send Physiological Data | 15 / day * #nPatient | low | < 1 second | C3 |
| ADUC 7: View Therapeutic Plan | 1 / day * #nPatient | low | < 30 minutes | C3 |
| ADUC 8: Send Diagnostic Information | 6000 / day | low | < 30 minutes | C4 |
| ADUC 9: Send Patient Health Information | 6000 / day | low | < 30 minutes | C4 |
| ADUC 10: Define Therapeutic Plan | 2 / hours * #nAttentingPhysician | medium | < 30 minutes | C5 |
| ADUC 11: Modify Therapeutic Plan | 2 / hours * #nAttentingPhysician | medium | < 30 minutes | C5 |
| ADUC 12: View Medical Records | 8 / day * #nAttentingPhysician | low | < 30 minutes | C5 |
| ADUC 13: Update Sensor Frequencies | 16 / hours * #nAttentingPhysician | low | < 5 minutes | C5 |
| ADUC 14: Confirm Phone Call | 16000 / day | low | < 30 minutes | C5 |
| ADUC 15: Patient Data Handler | 1 / 80 minutes + 1 / 3 minutes + 15 / day * #nPatient | high | < 1 minutes | C6 |
| ADUC 16: Verify Exercise Compliance | 1 / day | low | < 30 minutes | C7 |

# Solution A. Molteplicity and Comment

Each component is responsible for a specific set of related activities, thereby maximizing cohesion and minimizing inter-component dependencies.
Multiplicity is chosen based on component frequency and design constraints.

| Component | Multiplicity | Comment |
|---|---|---|
| C1 | #nPatient/700 (~100 instantiated components) | Sensors and Data Handler |
| C2 | 2 * #nHospital (one hot spare) | Alerts Handler |
| C3 | 1 * #nHospital/2 (only biggerHospitals) | Patient Handler |
| C4 | 1 * #nHospital/2 (only biggerHospitals) | On-Call Physician Handler |
| C5 | 1 * #nHospital/2 (only biggerHospitals) | Attenting Physician Handler |
| C6 | 2 * #nHospital (one hot spare) | Patient Data Handler |
| C7 | 1 * #nHospital/2 | Exercise compliance handler |

# Solution A. Evaluation per Component

C1: Medium values because it groups basic acquisition tasks, repeated frequently but with low complexity.

C2: Highest scores overall, since it manages hospital identification, notifications, and alerts, which are frequent and critical processes.

C3: Moderate values, as it handles viewing and sending data for patients, with longer interactions but lower intensity.

C4: Relatively low values, reflecting the on call physician functions'.

C5: High scores due to therapeutic plan management, which involves frequent updates, attending physician actors and longer interactions.

C6: Strong intra-component and sharing values because it aggregates patient data from multiple sources and processes them.

C7: Lowest overall, since it only verifies exercise compliance, which is a simple and daily recurrent task.

| Component | Abstraction | Complexity | Frequency | Delay | Location | Extra Flow | Intra Flow | Sharing | Control Flow |
|---|---|---|---|---|---|---|---|---|---|
| C1 | 20 | 33 | 22 | 25 | 25 | 15 | 15 | 50 | 20 |
| C2 | 40 | 100 | 33 | 25 | 75 | 60 | 0 | 70 | 0 |
| C3 | 20 | 33 | 22 | 50 | 25 | 15 | 15 | 50 | 10 |
| C4 | 20 | 33 | 11 | 25 | 25 | 15 | 0 | 70 | 0 |
| C5 | 20 | 66 | 44 | 50 | 50 | 30 | 0 | 70 | 20 |
| C6 | 20 | 33 | 11 | 25 | 0 | 0 | 15 | 90 | 50 |
| C7 | 20 | 33 | 11 | 25 | 0 | 15 | 15 | 20 | 0 |

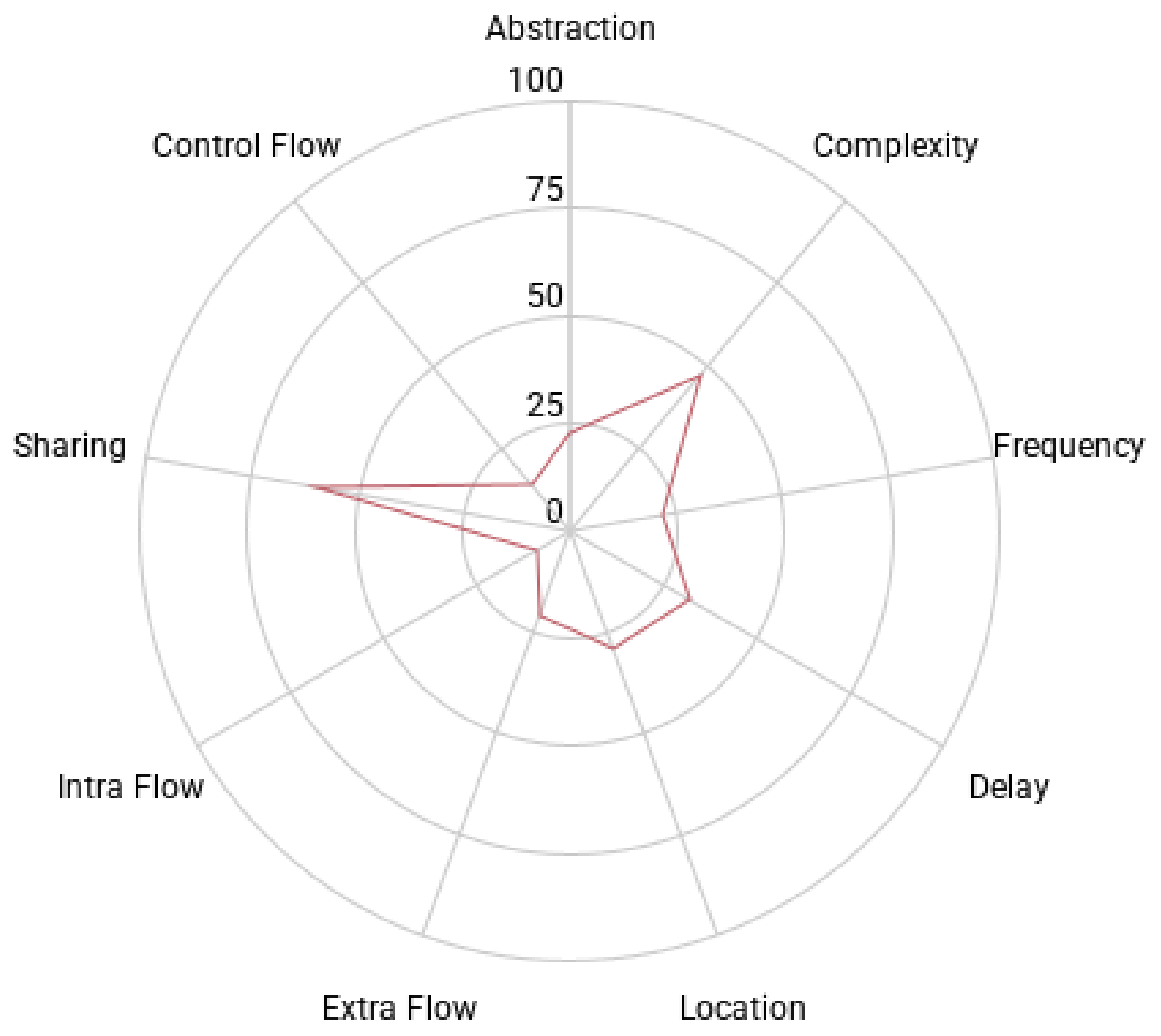
# Solution A.  Final Evalution

| Partitioning method | Abstraction | Complexity | Frequency | Delay | Location | Extra Flow | Intra Flow | Sharing | Control Flow |
|---|---|---|---|---|---|---|---|---|---|
| Functional | 23 | 47 | 22 | 32 | 29 | 21 | 9 | 60 | 14 |

The functional partitioning shows moderate abstraction and complexity, with low execution frequency and acceptable delays.

Data and processes are fairly localized, with limited extra and intra flows.

Resource sharing is high, while control flow remains relatively low, ensuring balanced coordination across components.



# Solution B. Activities

Technical Layers divides the system into technical layers (Data, Model, Presentation), each responsible for a technological aspect across all functionalities. This solution reduces sharing but complexity raises.

| Activity | Frequency | Complexity | Delay | Component |
|---|---|---|---|---|
| ADUC 1: Acquire Physiological Data | 1 / 3 minutes * #nPatient | low | < 1 second | C3 |
| ADUC 2: Acquire Daily Activity Data | 1 / 80 minutes * #nPatient | low | < 1 second | C3 |
| ADUC 3: Identify Hospital | 700 / day | high | < 2 minutes | C2 |
| ADUC 4: Identify Oncall and Notify | 6000 / day | medium | < 2 minutes | C2 |
| ADUC 5: Send Threshold Alert | 16000 / day | low | < 2 minutes | C2 |
| ADUC 6: Send Physiological Data | 15 / day * #nPatient | low | < 1 second | C1 |
| ADUC 7: View Therapeutic Plan | 1 / day * #nPatient | low | < 30 minutes | C1 |
| ADUC 8: Send Diagnostic Information | 6000 / day | low | < 30 minutes | C1 |
| ADUC 9: Send Patient Health Information | 6000 / day | low | < 30 minutes | C1 |
| ADUC 10: Define Therapeutic Plan | 2 / hours * #nAttentingPhysician | medium | < 30 minutes | C2 |
| ADUC 11: Modify Therapeutic Plan | 2 / hours * #nAttentingPhysician | medium | < 30 minutes | C2 |
| ADUC 12: View Medical Records | 8 / day * #nAttentingPhysician | low | < 30 minutes | C1 |
| ADUC 13: Update Sensor Frequencies | 16 / hours * #nAttentingPhysician | low | < 5 minutes | C2 |
| ADUC 14: Confirm Phone Call | 16000 / day | low | < 30 minutes | C1 |
| ADUC 15: Patient Data Handler | 1 / 80 minutes + 1 / 3 minutes + 15 / day * #nPatient | high | < 1 minutes | C3 |
| ADUC 16: Verify Exercise Compliance | 1 / day | low | < 30 minutes | C2 |

# Solution B. Evaluation per Component

C1: High abstraction and delay reflect the user interface responsibilities, with moderate complexity and sharing.

C2: Very high complexity and location values indicate centralized processing of business rules, while low control flow and intra-flow suggest lower interaction with other components.

C3: Lower abstraction and frequency but higher intra-flow and control flow reflect data management responsibilities and interactions with other layers.

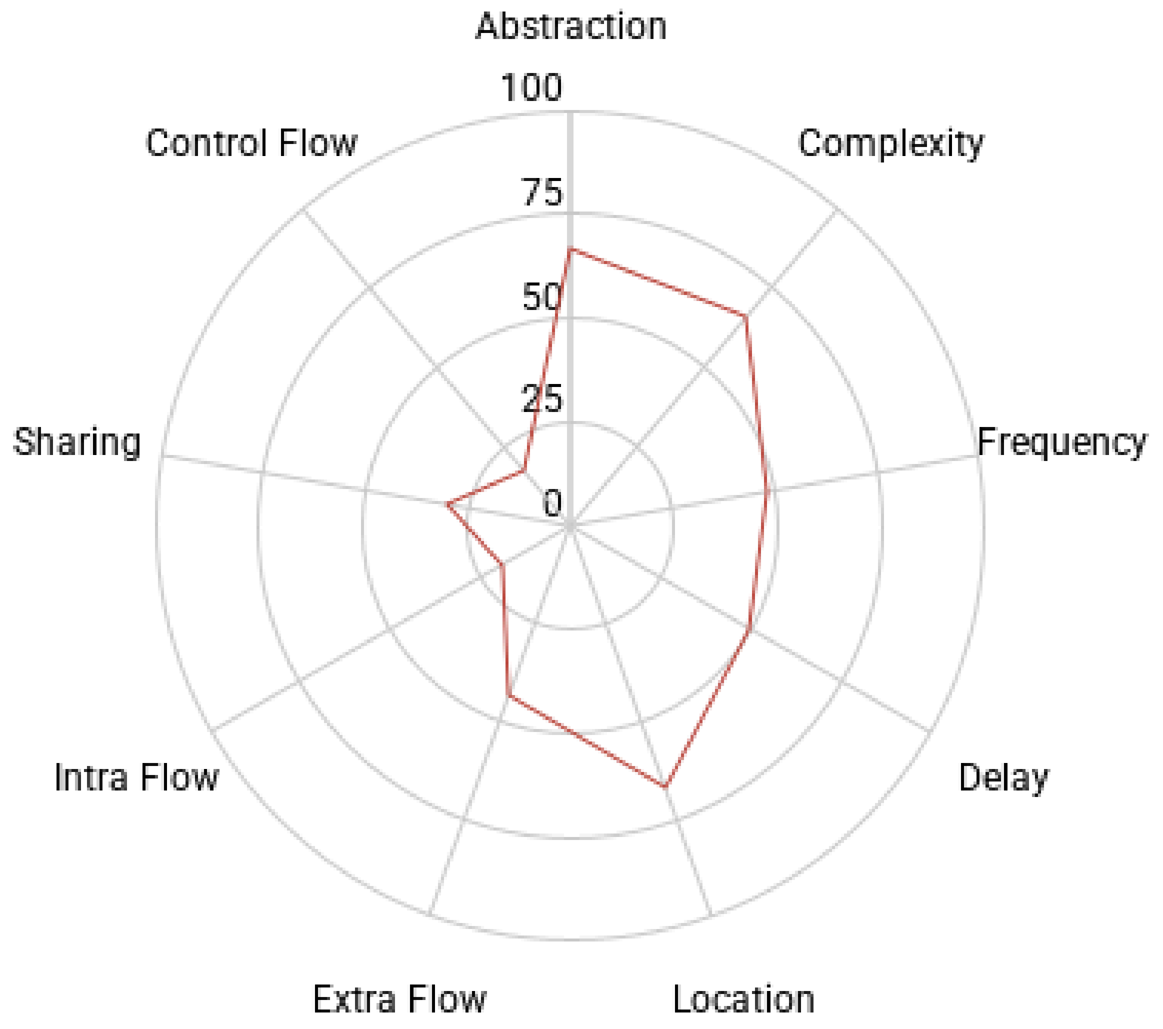
| Component | Multiplicity | Comment |
|---|---|---|
| C1 | 1 *nHospital | View Plane |
| C2 | 1 *nHospital | Logic Plane |
| C3 | 1 *nHospital | Data Plane |

| Component | Abstraction | Complexity | Frequency | Delay | Location | Extra Flow | Intra Flow | Sharing | Control Flow |
|---|---|---|---|---|---|---|---|---|---|
| C1 | 80 | 33 | 55 | 50 | 75 | 57 | 8 | 50 | 20 |
| C2 | 85 | 100 | 55 | 50 | 100 | 42 | 0 | 30 | 0 |
| C3 | 35 | 66 | 33 | 50 | 25 | 29 | 50 | 10 | 30 |

# Solution B.  Final Evalution

| Partitioning method | Abstraction | Complexity | Frequency | Delay | Location | Extra Flow | Intra Flow | Sharing | Control Flow |
|---|---|---|---|---|---|---|---|---|---|
| Technical Layers | 67 | 66 | 48 | 50 | 67 | 43 | 19 | 30 | 17 |

The metrics confirm that Technical Layers increase coupling and reduce cohesion compared to functional partitioning.

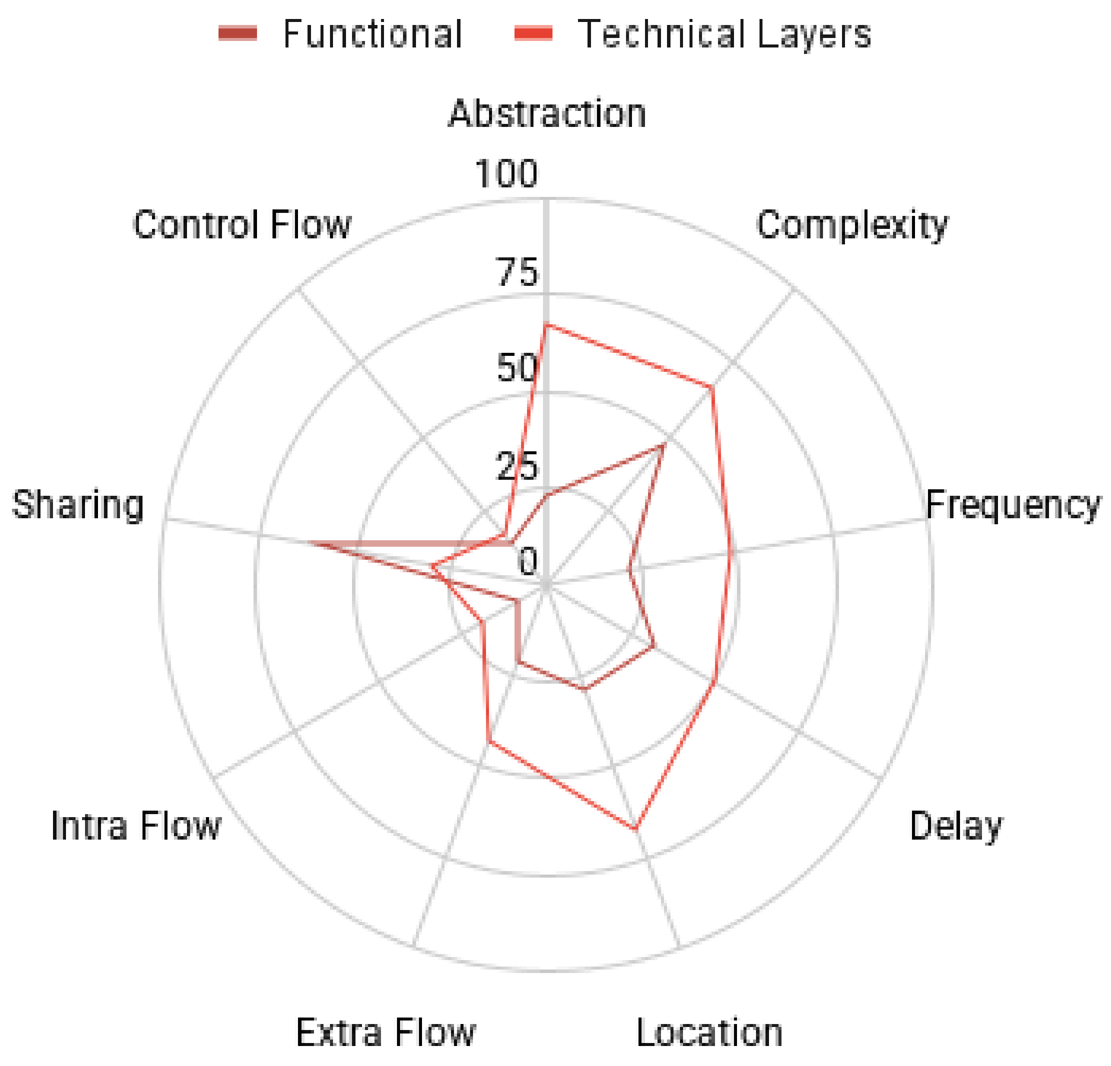


# Final Comparison

| Partitioning method | Abstraction | Complexity | Frequency | Delay | Location | Extra Flow | Intra Flow | Sharing | Control Flow |
|---|---|---|---|---|---|---|---|---|---|
| Functional | 23 | 47 | 22 | 32 | 29 | 21 | 9 | 60 | 14 |
| Technical Layers | 67 | 66 | 48 | 50 | 67 | 43 | 19 | 30 | 17 |

Functional partitioning shows lower average values (Abstraction 23 vs 67, Complexity 47 vs 66, Frequency 22 vs 48, etc.), indicating smaller area, better cohesion, and reduced coupling, whereas Technical Layers result in higher values across most metrics.

We choose Functional Partitioning because it yields a more compact design with higher cohesion and lower coupling, making it the preferable approach.

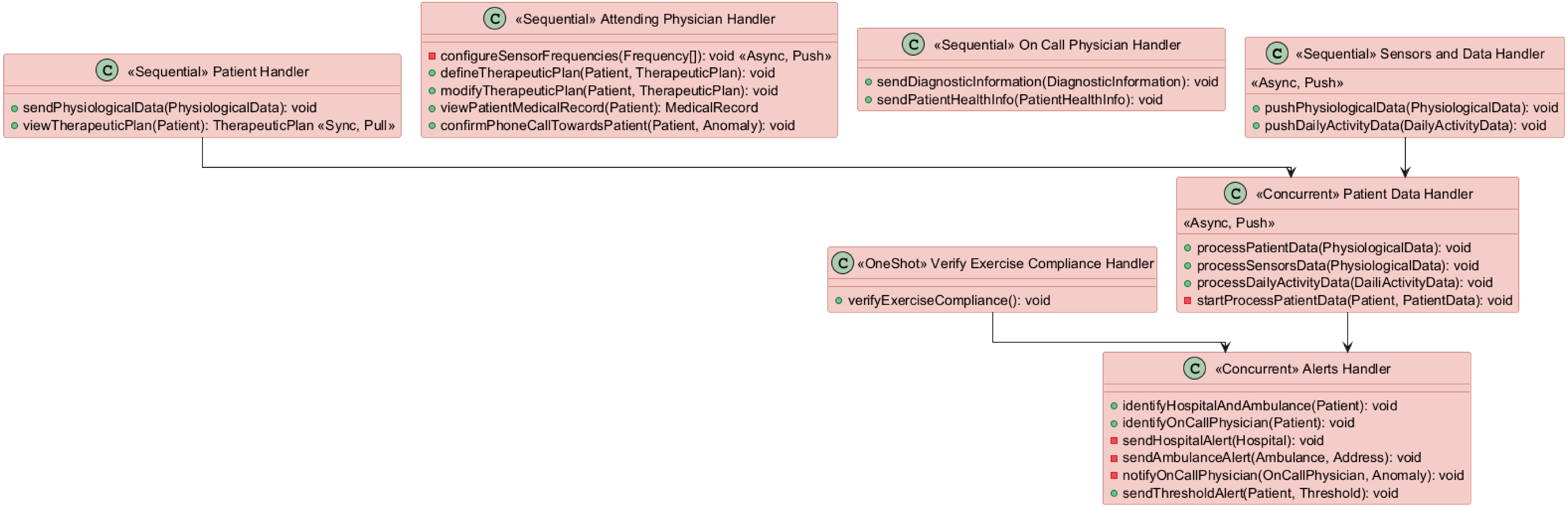


# Concrete Architecture

# 06

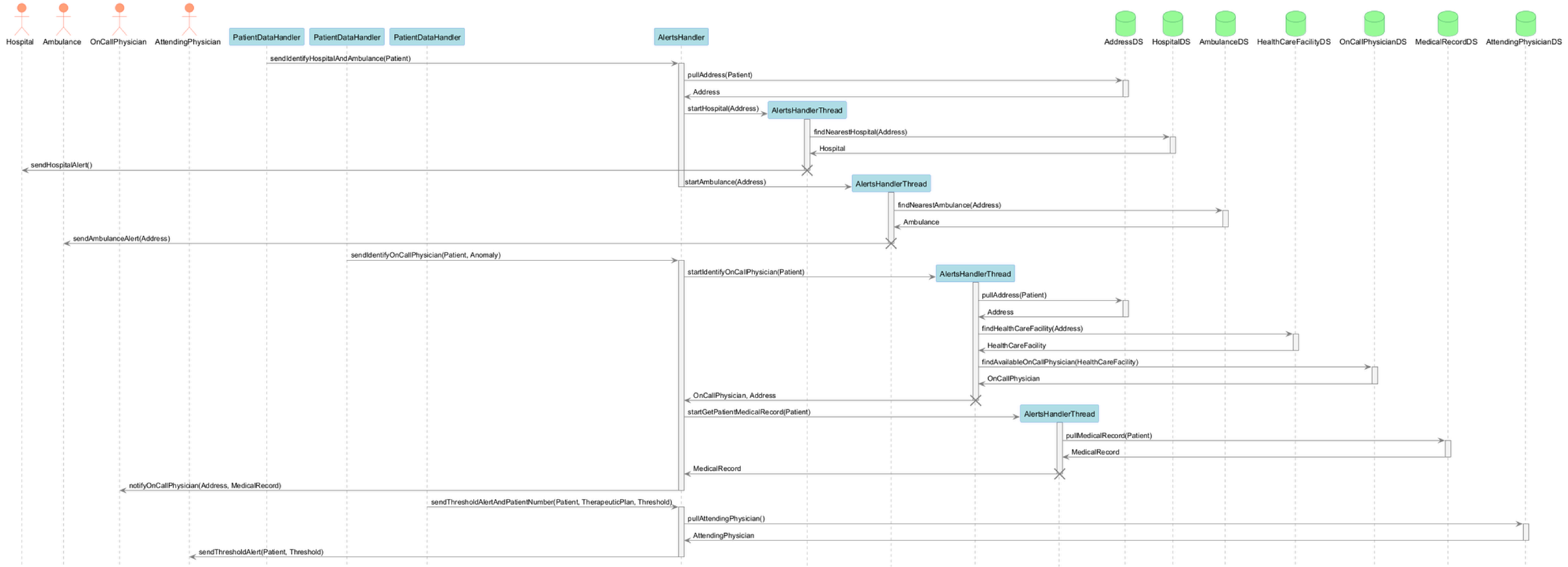
# Concrete Diagram

All components are stateless and active; as a result, they can be easily scaled horizontally, replaced in case of failure, and maintained without disrupting the overall system. This design choice increases resilience, simplifies fault recovery, and supports high availability



# Alerts Handler

The Alert Handler component manages all alerts. When it receives an event from the PatientDataHandler, it processes all events concurrently according to their alert level, and then elaborates them accordingly.



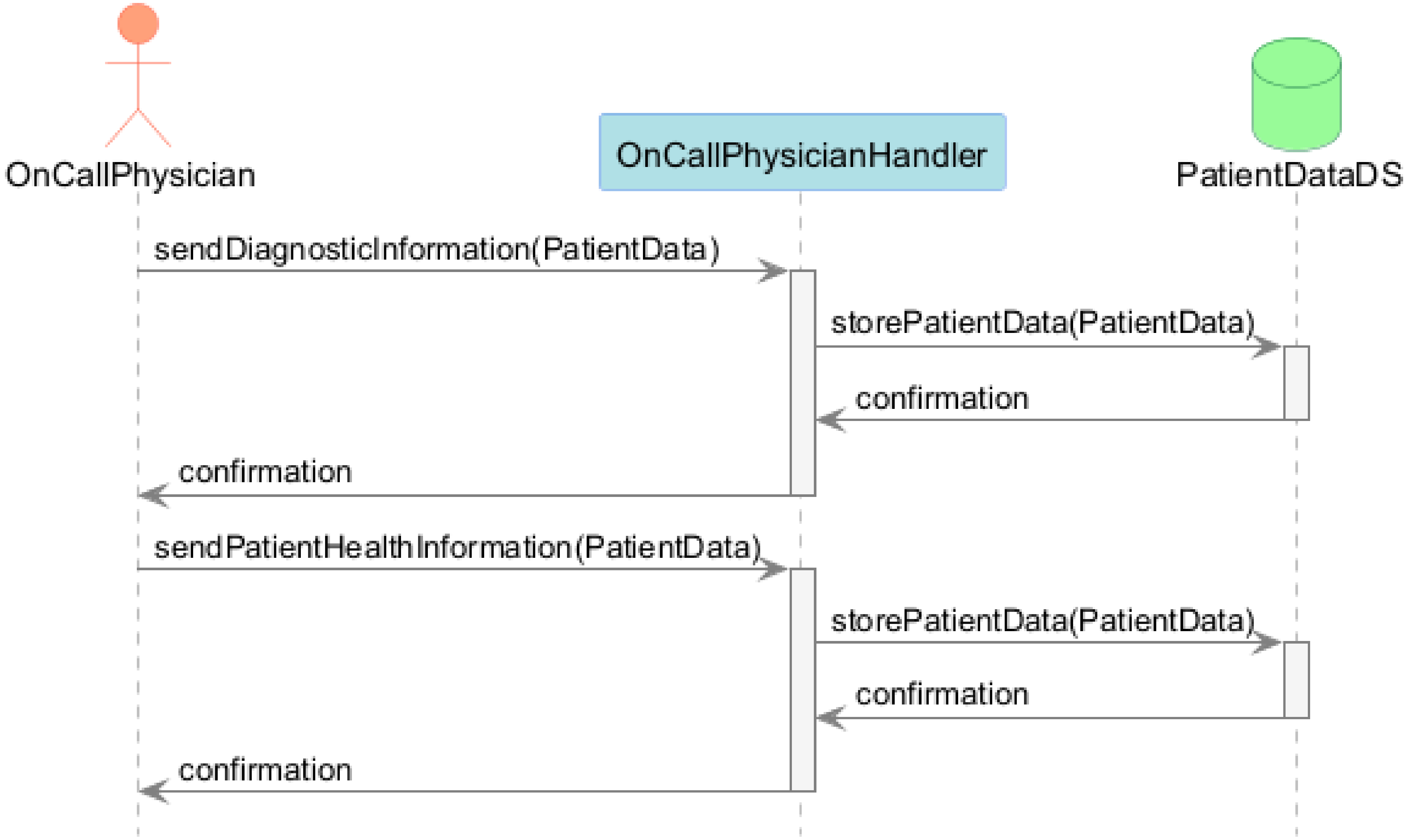
# Attending Physician Handler


The AttendingPhysicianHandler manages all actions performed by the attending physician, which are generally less critical. These actions are executed sequentially.

06

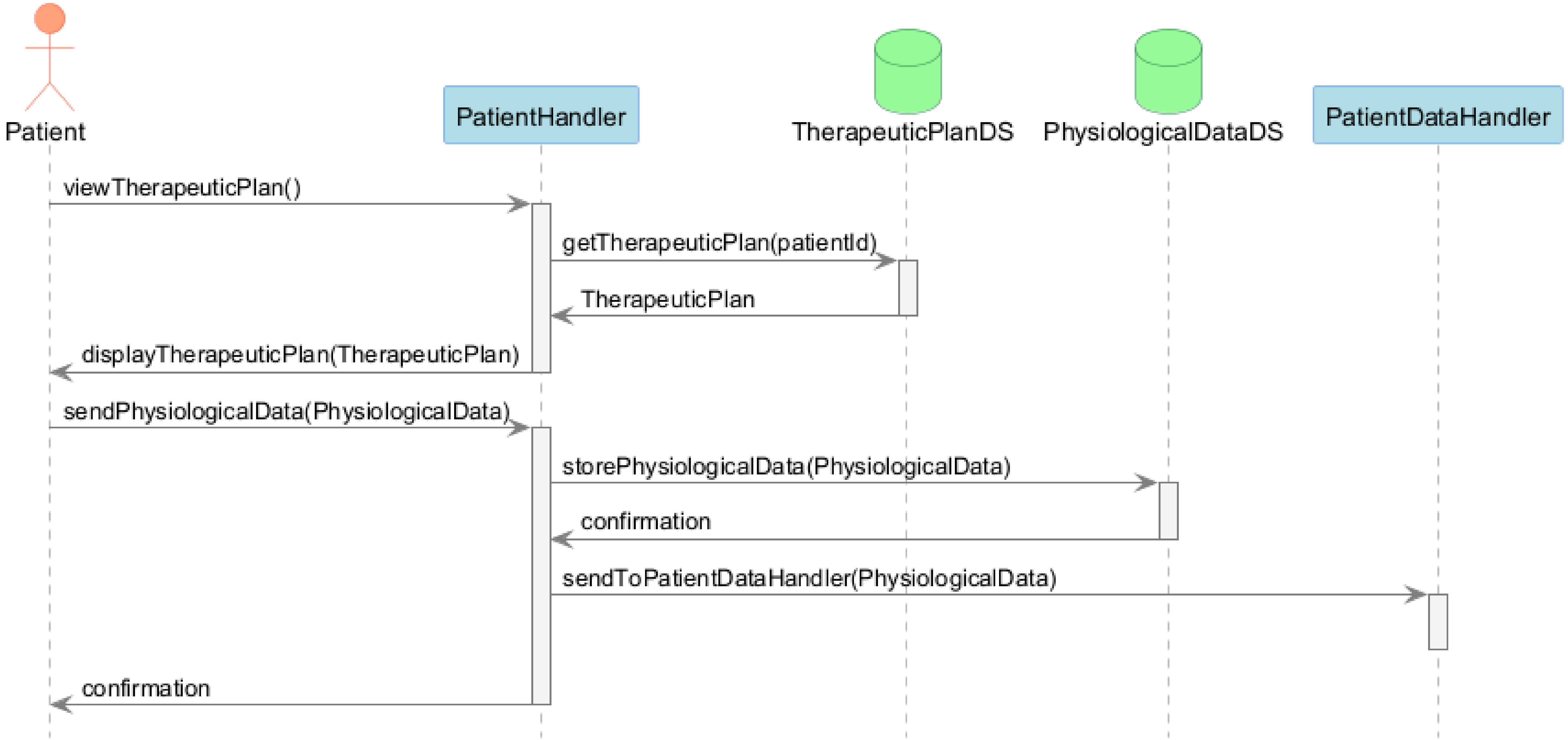
# On Call Physician Handler

The OnCallPhysicianHandler performs the same role as the AttendingPhysicianHandler, with the difference that it operates when an OnCallPhysician send diagnostic or patient health information.

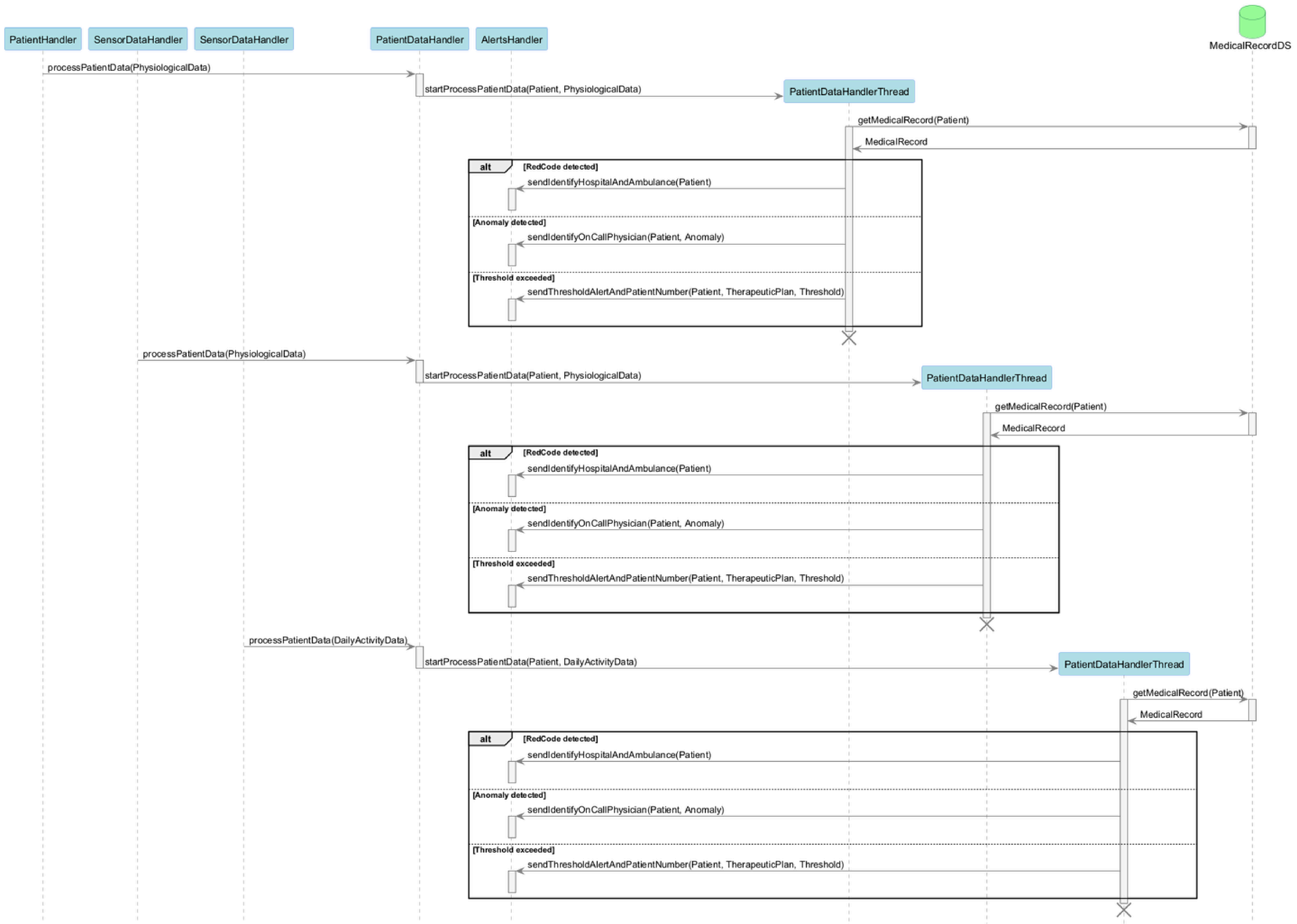


# Patient Handler

The Patient Handler manages patient actions and interacts with the Data Processing component.



# Patient Data Handler

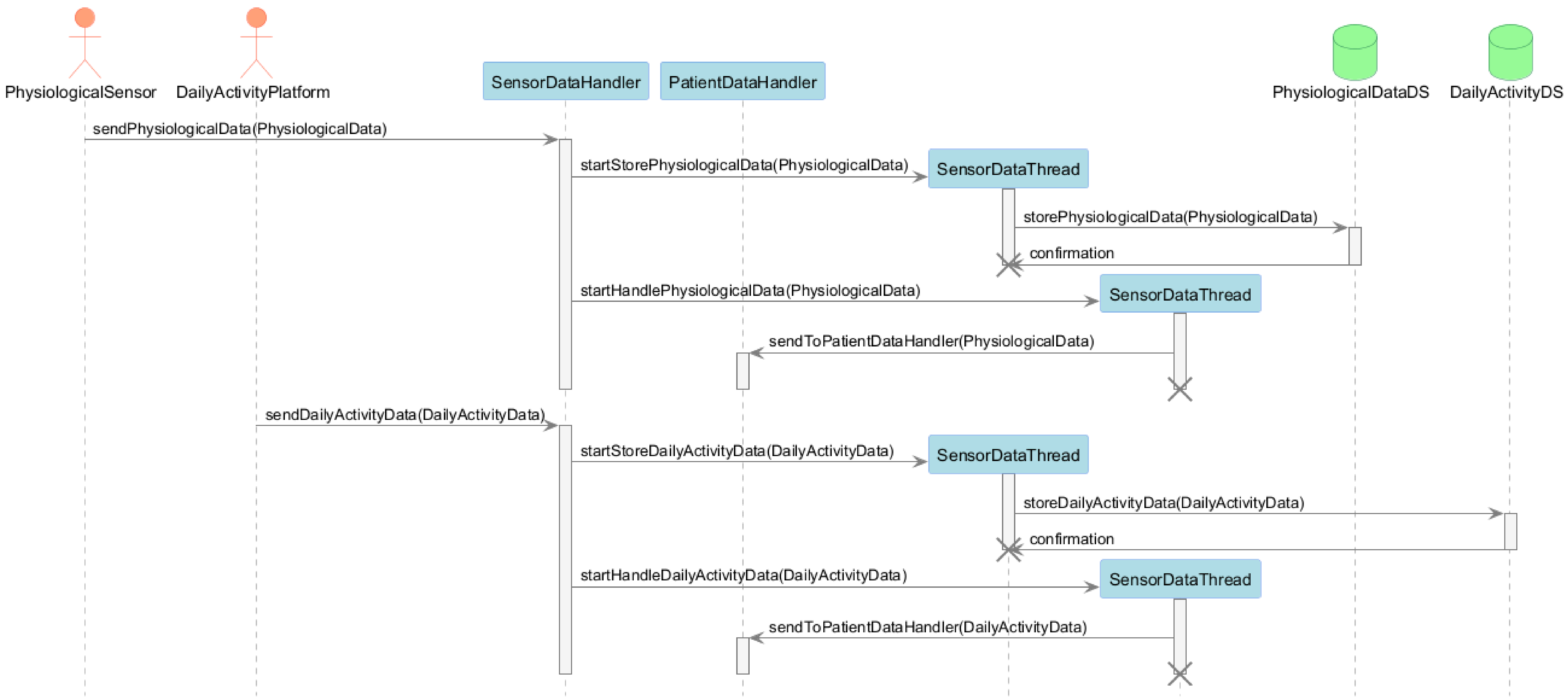



The PatientDataHandler is the main component responsible for concurrently managing data from different sources. After analyzing the data to detect anomalies, threshold exceeded or not reached and red code, it triggers alert events and dispatches them to initiate the corresponding actions.

06

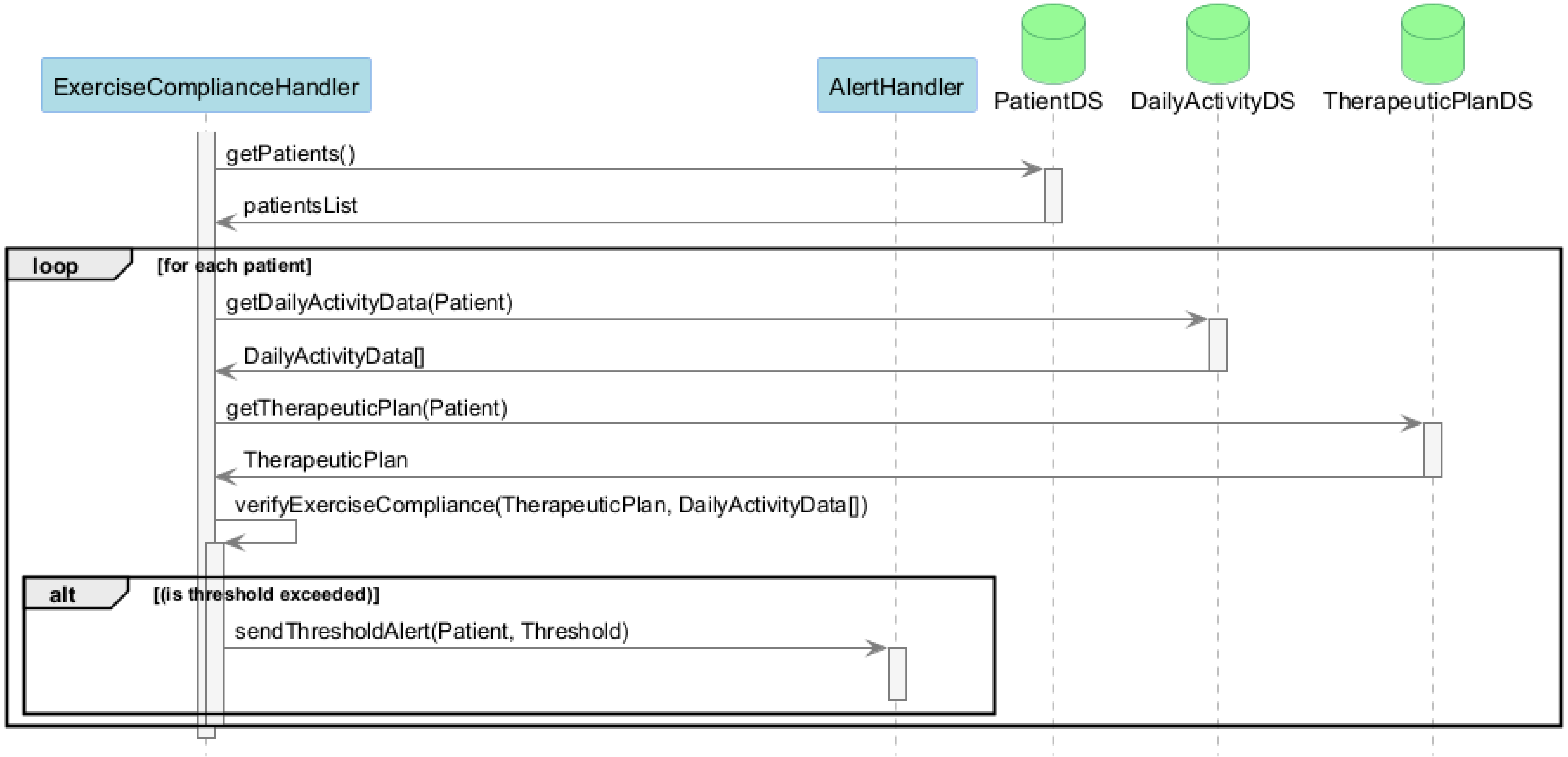
# Sensors Data Handler

The SensorDataHandler manages data collected from sensors, storing it and also forwarding it to the PatientDataHandler to be processed as events.



# Verify Exercise Compliance Handler

The ExerciseComplianceHandler periodically checks whether patients have completed the activities in their therapeutic plan. If not, it triggers a 'threshold exceeded' event to notify the attending physician.

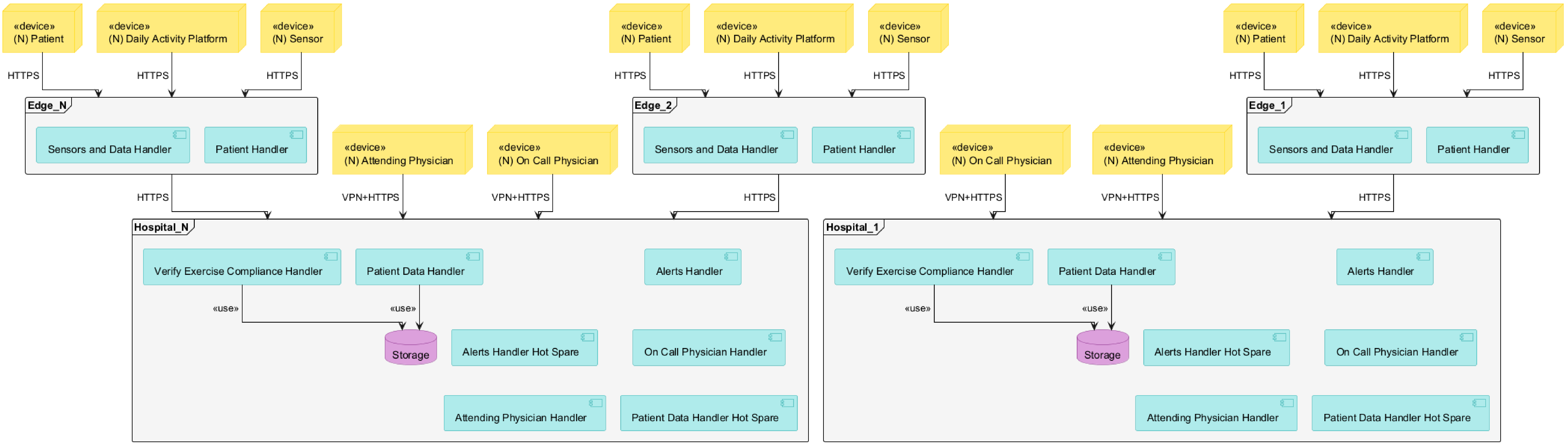


# Deployment Architecture

07

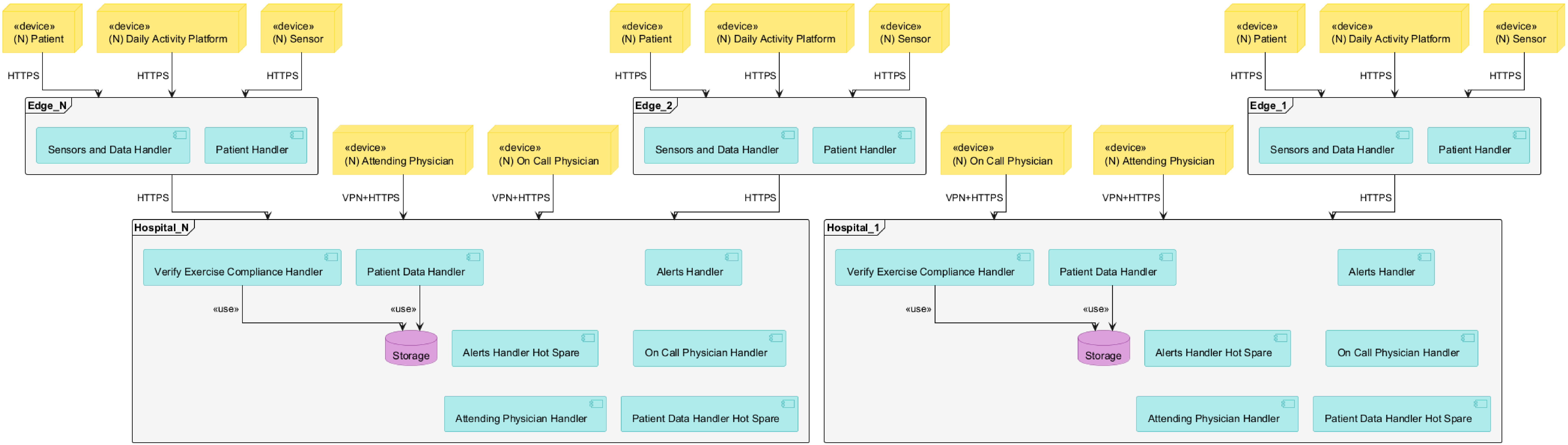
# Deployment Architecture 1

The system is designed to be distributed across hospitals, allowing each to manage local operations autonomously. This aligns with the expectation that areas with more patients will have more doctors. Additionally, edge nodes handle sensor data and patient operations, as their proximity enables faster data transmission and processing. Costs are significantly lower compared to building a new system, as it leverages existing infrastructure. Hot Spare increases availability and performance by allowing rapid recovery in case of component failure.



# Deployment Architecture 2

The figure shows a simplified example with N hospitals using shared storage to ensure redundancy, allowing continuous data processing and improved availability. Not all hospitals will host all components; for instance, less critical actions, such as those handled by the On-Call Physician Handler, can be managed by nearby hospitals. All main communications are conducted over HTTPS, ensuring encryption and preventing data leakage. For highly sensitive data, such as those managed by the Attending and On-Call Physician components, a VPN is required to provide enhanced security.



# Solution Evaluation

# 08

# Pros

Fault Tolerance: Redundant and stateless components ensure that failures in one part of the system do not compromise the overall service, increasing robustness and reliability.

Availability: The distributed architecture ensures continuous operation even if some nodes fail, thanks to redundancy and shared storage.

Security: Critical communications are encrypted (HTTPS) and sensitive components use a VPN, protecting data from leaks and unauthorized access.

Deployability: Components can be deployed independently across hospitals and edge nodes, simplifying scaling and maintenance.

Modifiability: Updates or changes can be applied to individual components without affecting the entire system, supporting easier upgrades and adaptations

Low Latency: The connection between edge nodes and hospitals is fast, as the network infrastructure is typically robust, enabling quicker data transmission and processing.

# Cons

Increased Complexity: Managing multiple hospitals and edge nodes requires sophisticated coordination and monitoring.

Latency Variability: While edge nodes improve local performance, inter-hospital communication may still experience delays.

Resource Heterogeneity: Not all hospitals host all components, which may lead to uneven load distribution and dependency on neighboring nodes.

Maintenance Overhead: Updating and patching distributed components can be more labor-intensive compared to a centralized system.

Security Management: Despite HTTPS and VPNs, managing credentials, certificates, and secure channels across multiple sites adds operational complexity.

# Future Developments

Extend the system to Elderly Patients: Extend the system for Elderly Patients allowing them to stay at home and still be monitored checking their health status.

Enhanced Predictive Analytics: Implement machine learning models to predict potential complications or non-adherence before thresholds are breached.

Automated Teleconsultations: Allow the system to initiate video consultations with the physician automatically when certain alerts are triggered.

Personalized Rehabilitation Plans: Use historical data and analytics to dynamically adjust therapeutic plans for each patient based on performance and recovery trends.

# Bibliography


1. M. della Salute. "Annuario Statistico del Servizio Sanitario Nazionale - Anno 2024." [Online]. Available: https://www.salute.gov.it/nssr/home/statistiche/annuario/annuario.jsp
2. "Real-Time Remote Patient Monitoring: A Review of Biosensors Integrated with Multi-Hop IoT Systems via Cloud Connectivity." Applied Sciences, vol. 14, no. 5, p. 1870, 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/5/1870
3. A. Nasr, "Implantable Medical Electronic Devices: Sensing Mechanism, Communication Methods and the Future." Journal Name, 2024.
4. "Remote patient monitoring systems: Applications, architecture, and challenges," Science Direct, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468227623000959
5. "BETTER HOMO2 Accessed: Aug. 17, 2025. [Online]. Available: https://www.better-health-project.eu/

# Questions?

Thanks for the attention!