



# Forward Deployed Engineer

## Technical Challenge

### Overview

At Lyrebird, our Forward Deployed Engineers (FDEs) work closely with customer systems, integrating our platform with clinical software.

This challenge tests your ability to build, reason about, and communicate a simple integration that emulates the kind of work we do every day - connecting a healthcare system using HL7 with an HTTP-based API.

### The Challenge

You will build a local HL7 message sender and receiver that:

1. Create a HL7 publisher to send messages to your receiver (simulate an upstream system).
2. Receiver should listen for incoming HL7 messages over TCP (e.g., MLLP protocol).
3. Securely parse and transform the HL7 message into a JSON payload.
4. Send that JSON payload to a sample HTTPS REST API endpoint (you can stub one locally).
5. Demonstrates some error handling and provides limited test cases.

### Core Requirements

#### 1. HL7 Listener and Sender

- Implement a lightweight HL7 receiver that listens on a local TCP port (e.g., port 2575).
- Implement a sender that can publish an HL7 message to that listener (e.g., a simple ADT^A01 message).
- Use the **MLLP (Minimal Lower Layer Protocol)** for framing.

#### 2. Message Transformation

- Parse the HL7 message into a structured JSON format.
- Map key fields (e.g., patient ID, name, DOB, message type).
- Send this JSON payload via POST to a local HTTP API endpoint (e.g., <http://localhost:8080/api/v1/messages>).



### 3. Clarity and Design

- Organize the project with clear structure and naming.
  - Include a short **README.md** that explains:
    - How to run it locally (using Docker, Python, Node, Go — your choice).
    - How it works.
    - What trade-offs you made.
- Ideas for improving reliability in production.

## Deliverables

- A GitHub repo (or zipped folder) containing:
  - Source code
  - **README .md** with setup instructions
  - Example HL7 message and sample API output

