1. Introduction

1.1 Slide 1: Problem & The Sentinel Solution

"Good afternoon, Dr. Mellah and fellow students. My name is Massimo Caruso, and today I am presenting my final project for SOEN 422: The **Secure LoRa Asset Monitoring System**, or as I call it, the 'Sentinel.'

The core problem I addressed is the vulnerability of high-value university assets. Standard key locks offer no audit trails, and modern Wi-Fi security systems have a critical weakness: if the campus network goes down, or if power is cut, the security system goes blind.

My proposed solution is an **'Air-Gapped' Security System**. It operates on an independent, long-range LoRa network that doesn't rely on Wi-Fi. It combines smart physical locking with dual-layer authentication—using mobile Bluetooth for the user and a secure biometric console for the admin.

---

2. System Architecture: The Monitor

2.1 Slide 2: Architecture - Monitor Node

Let's look at the heart of the system: The **Monitor Node**, powered by a TTGO LoRa32 T3. This device is mounted on the asset door.

This isn't just a simple loop; it runs on a robust **Finite State Machine**. When the system is **Disarmed**, the servo motor actually follows the user's hand movements on the potentiometer, allowing for manual door control.

However, once I send the **ARM** command via Bluetooth, the system enters a specific **'Locking State.'** It drives the servo strictly to 0 degrees and calibrates the sensors.

A major technical challenge I solved here was signal noise. The potentiometer would jitter, causing false alarms. To fix this, I implemented **Deadband Filtering**—ignoring changes smaller than 30 units—and signal averaging. Now, the alarm *only* triggers on a deliberate 'Forced Entry.'

If that threshold is breached, the system locks down, flashes a Red LED, and broadcasts an encrypted alert. The only way to unlock it is through a **Bluetooth Challenge-Response protocol**. The system sends a random numerical challenge to the phone, and the user must respond with the correct key.

---

3. System Architecture: The Gateway

3.1 Slide 3: Architecture - Gateway Node

On the receiving end, we have the **Gateway Node**. This acts as the central security hub.

During development, I made a critical engineering decision to move away from a web-based interface, which proved unstable with the radio stack, to a robust **Serial Admin Console**.

This console implements strict **Defense in Depth**:

1. First, it receives the AES-128 encrypted LoRa packets.
2. To view the decrypted data, an Admin must log in.
3. I implemented a **Three-Factor Authentication** simulation here. You must enter a username, a password, *and* simultaneously hold a physical **Biometric Touch Sensor** (Pin 13).

If any of these fail three times, the code triggers a **3-Strike Lockout**, freezing the system for 60 seconds to prevent brute-force attacks.

When a valid 'Forced Entry' packet is received and decrypted, the Gateway triggers an active buzzer on Pin 21 to alert security personnel immediately.

---

4. Design Assessment & Reflections

4.1 Slide 4: Design Assessment

Finally, I'd like to assess the design regarding reliability, security, and ethics.

For Reliability: The biggest success was the implementation of **Non-Blocking Logic**. By using millis() timers instead of delay(), the servo can move, the LED can blink, and the LoRa radio can transmit simultaneously without hanging the CPU.

For Security: We used **AES-128 encryption** for all over-the-air traffic. Furthermore, the decision to decouple the 2FA logic from the tamper check in the code prevents the system from entering an infinite loop during an attack, ensuring the user can always attempt to disarm.

For Safety: I implemented a **Fail-Safe design**. On power-up or reset, the system defaults to the 'Disarmed' state with potentiometer control enabled. This ensures that if the logic fails, the user isn't permanently locked out of their own asset.

Ethically: The biometric system respects privacy. Instead of storing actual fingerprint images, we rely on local capacitive threshold values, ensuring no personal biological data is ever stored or transmitted.

---

5. Conclusion

In conclusion, this project demonstrates that we can build a high-security, responsive asset monitor using low-cost embedded hardware, provided we handle the concurrency and state management correctly.

Thank you for listening. I am happy to answer any questions about the code or the hardware."