

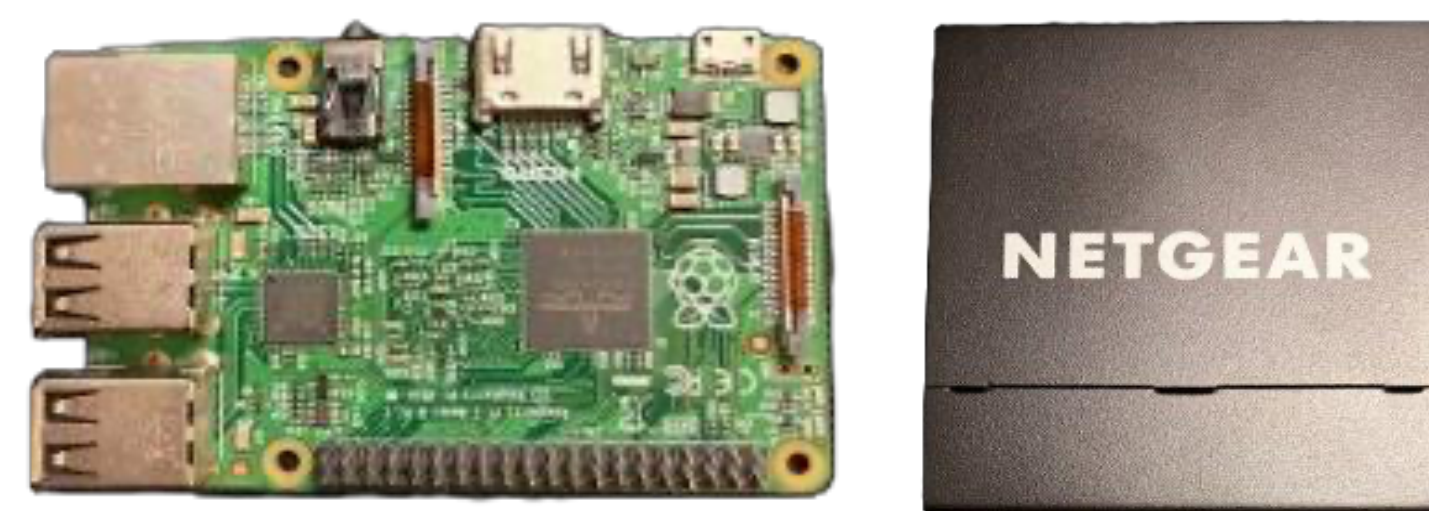
Motivation

Innoflight is a leading Non-traditional Defense Contactor (NDC) that delivers integrated avionics to the satellite market. These avionic devices require extensive testing to ensure reliability and safety. Test failures can occur due to issues in the testing environment, such as disconnected hardware or network problems. These environmental faults can waste engineering time and delay debugging.

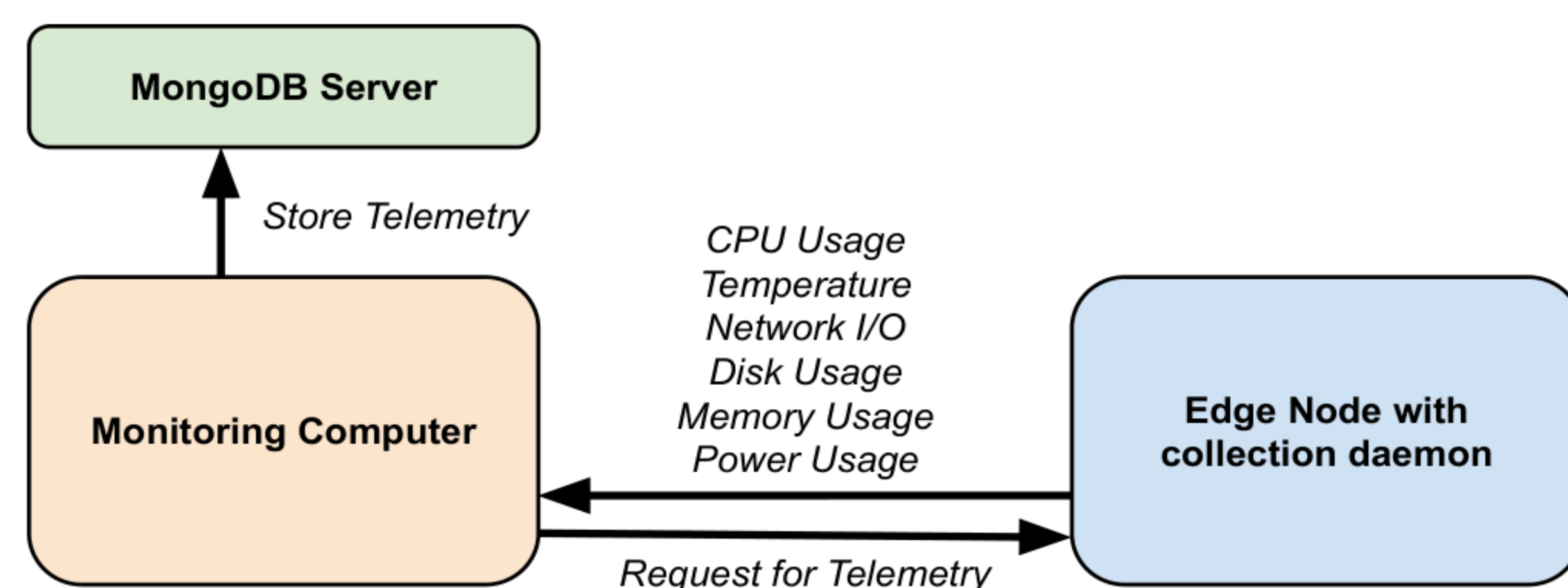
Solution: An automated and modular monitoring system that notifies engineers when a device is down or too slow during testing.

Device Test Environment

- Evaluated three deployment modes: Hardware, Simulation, and Hybrid.
- Simulation mode used Docker to emulate monitoring and edge nodes, enabling scalable and configurable testing.
- Hybrid mode combined Docker-simulated nodes with physical hardware for scalable testing in a realistic environment.



Advanced Telemetry Collection

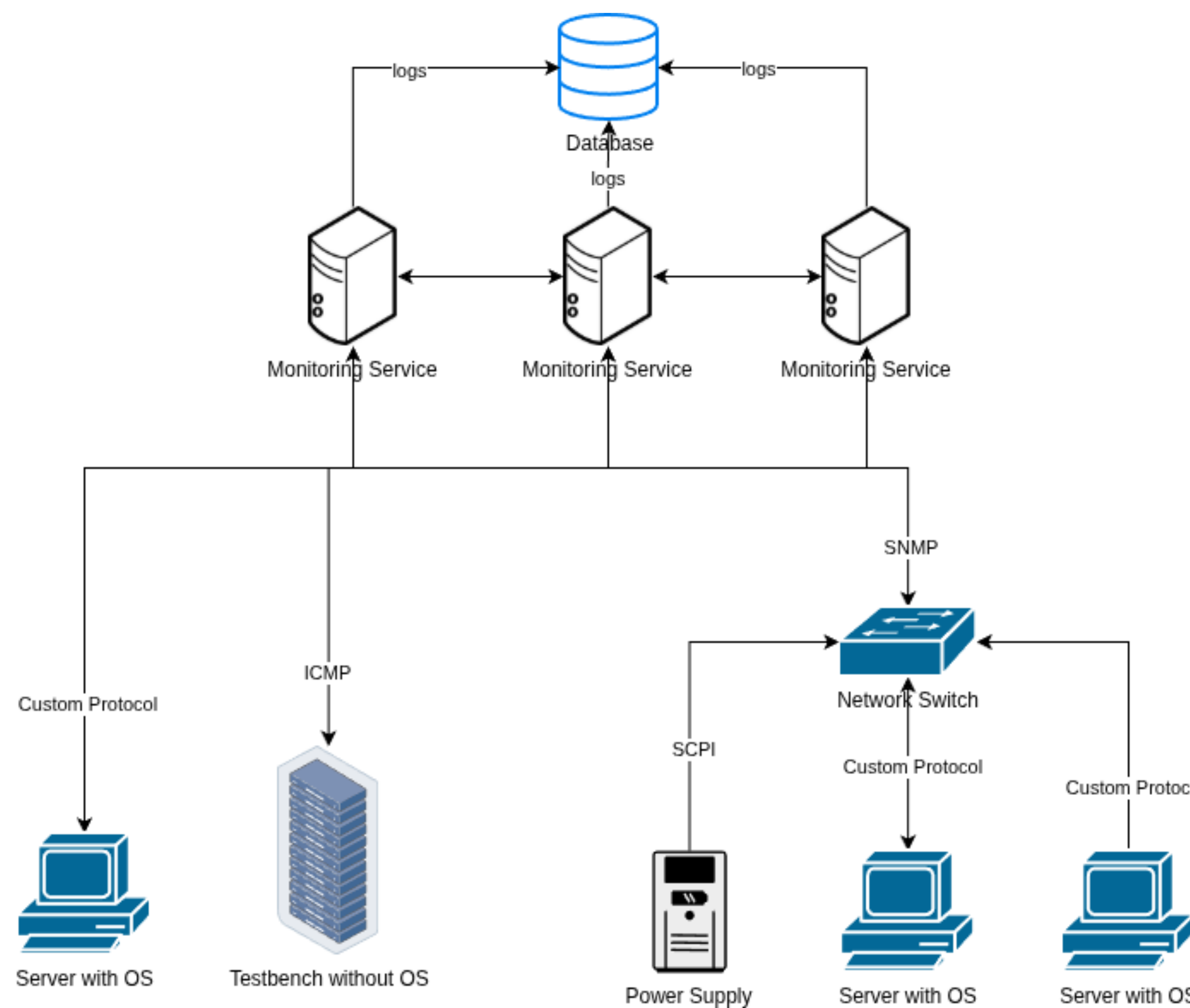


- The monitoring node periodically requests telemetry from each edge node over the network.
- The specific metrics collected can be tailored per device deployment, allowing for devices to report the telemetry that they are capable of collecting.
- Collected telemetry is written to the database and can be used for real-time device monitoring and post-outage diagnostics.

Fault Detection & Alerting

- Faults occur when the latency reported for an edge node exceeds the configured threshold or the device becomes unresponsive.
- An email alert is sent that includes the device information, latency, severity, and additional telemetry data depending on the alert configuration.

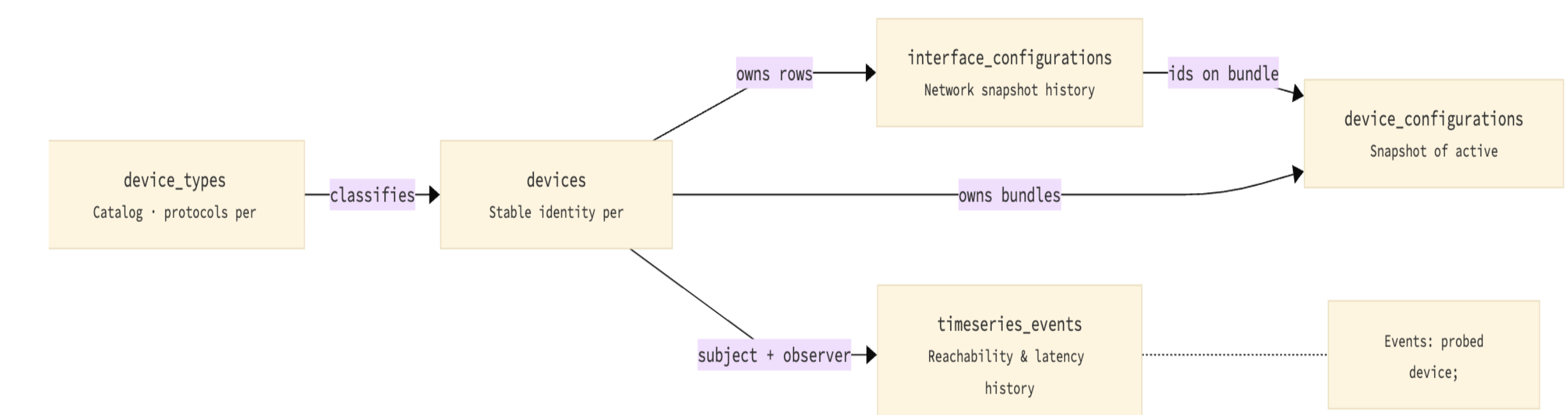
Monitoring Overview & System Architecture



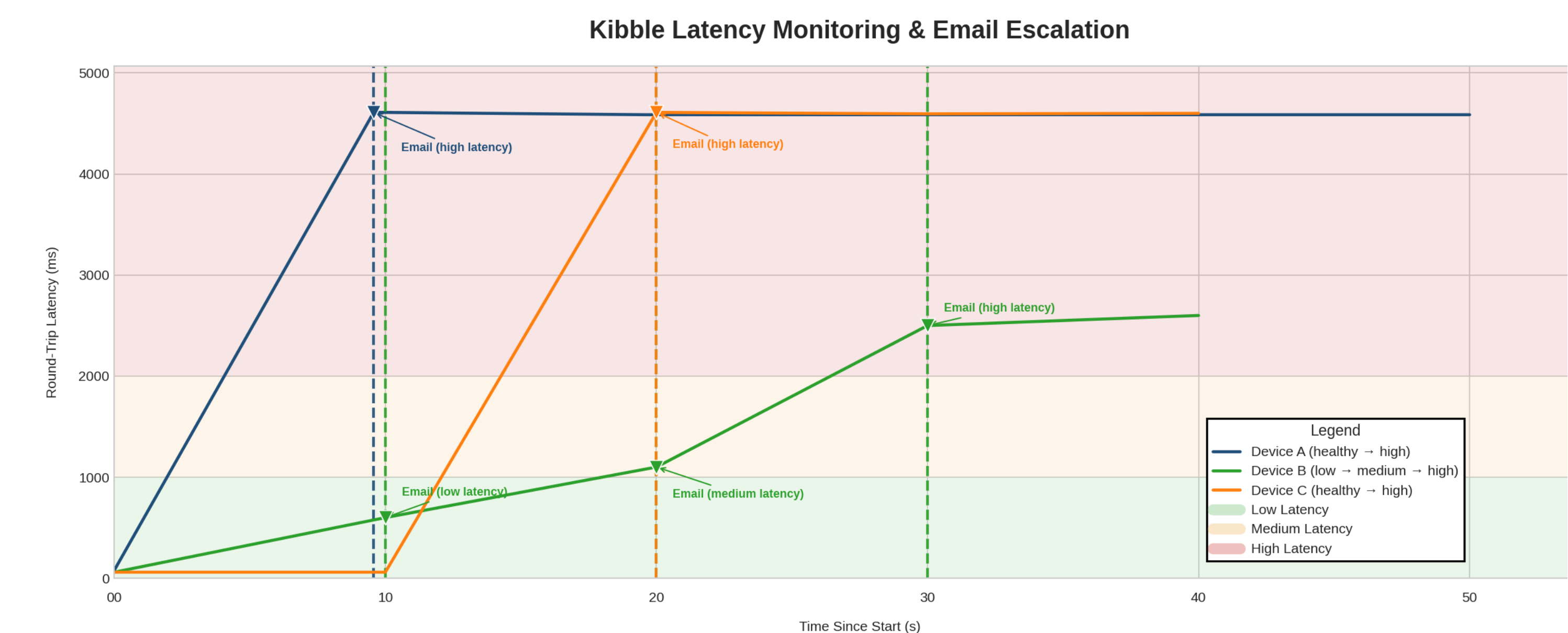
Example Network Configuration

- The central database stores all logs generated by the monitoring service computers
- The system supports multiple types of devices and logs generated
- Basic test devices may only have SCPI enabled, some more advanced may only have ICMP, full-fledged servers can run a custom protocol that reports complex telemetry, and network switches can report the status of the network using SNMP

MongoDB Schema & Data Logging



Database Schema Flow



Simulated Latency Monitoring Example

Distributed Monitoring Network

- Multiple monitoring nodes utilize MongoDB as a single source of truth
- Leader election process: each monitor writes a heartbeat to a shared collection and the monitor with the lowest ID that has a fresh heartbeat is elected leader
- All monitors probe all devices and log latency events. Only the leader sends alerts, which prevents duplicate notifications.
- If a monitor dies, its leases expire and surviving monitors pick up the orphaned devices automatically.
- The system stays operational even when a monitoring node goes down, with automatic failover in one heartbeat TTL (~30s).

Future Work

- Expand support for additional devices, communication protocols, telemetry collection, and alert types.
- Integrate graphing and visualization into the software stack to simplify monitoring and improve visibility.