

Master Documentation: Updates for EMsync

Project: Emergency Coordination & Management System

Lead Developer: Alan Joseph P V

Status: Active Development (Post-Phase 3 Refinement)

1. Core Infrastructure & Deployment

- **Orchestration:** The system is managed via **Docker Compose**, running three isolated microservices (ML, Routing, Transfer).
- **DevOps:** Automated **CI/CD** pipelines are triggered by GitHub Actions, building images to Docker Hub and maintaining a "Thin Client" local setup.
- **Network Protocol:** Services communicate over a private emsync-network bridge with host="0.0.0.0" bindings to ensure container-to-host accessibility.

2. Functional Module Refinements (Phase 3+)

A. Inter-Hospital Transfer (Secure Coordination)

- **Resource Expansion:** Migrated from a single "Bed" count to a high-granularity tracking system supporting **12 specific emergency categories** (e.g., Cardiac ICU, NICU, Dialysis, Ventilators).
- **Database Schema:** Updated Hospital and TransferSession models to store resource inventories as JSON-based dictionaries for dynamic scaling.
- **Real-Time Synchronization:** Implemented **Flask-SocketIO** broadcasts to ensure that accepting a transfer instantly decrements the specific resource count across all connected dashboards.

B. Security & Data Integrity

- **Encryption Bug Fix:** Resolved a data mapping error where the Date of Birth (DOB) was incorrectly encrypted.
- **JSON Handshake:** Identity data (Name, DOB) and the **Requested Service Type** are now bundled into a structured JSON payload before **AES-256 CBC** encryption, ensuring 100% data integrity during the transfer handshake.

3. UI/UX Enhancements (React Shell App)

- **Service Selection:** Integrated a dynamic dropdown in the **Registry (Step 1)** to allow paramedics to book specific emergency services alongside patient entry.
 - **Dynamic Admin View:** Developed a "drill-down" interface in **Admin View (Step 4)** where clicking a hospital opens a specific resource management panel that updates in real-time via WebSockets.
 - **System Reset:** Improved the **Reset Simulation** function to clear all active database sessions and reset all 12 resource categories to their default values across the entire network.
-

4. Technical Specification Summary

Component	Technology	Update Log
Backend	Flask / SQLAlchemy	JSON-based resource storage.
Security	PyCryptodome (AES-256)	Structured JSON payload encryption.
Frontend	React / Lucide-React	Dynamic Admin View & Hospital Selector.
Real-time	Socket.IO	Automated resource_update broadcasts.

```
C:\Windows\system32\cmd.e X + ▾

6 files changed, 315 insertions(+), 13 deletions(-)
create mode 100644 phase3-transfer\__pycache__\hosital_model.cpython-313.pyc

(virtual_network_emsync_env) G:\Final Year PROJECT\p4_final EMSYNC\emsync-root\phase3-transfer>git push origin main
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 16 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (11/11), 6.07 KiB | 327.00 KiB/s, done.
Total 11 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/Alan-JPV/EMSYNC_DOCKER_BACKEND.git
  5fba7f0..29509cb  main -> main

(virtual_network_emsync_env) G:\Final Year PROJECT\p4_final EMSYNC\emsync-root\phase3-transfer>python init_db.py
Database reset and initialized with 12 service categories for: ['Aster Medicity', 'Lourdes Hospital', 'Rajagiri Hospital', 'General Hospital Ernakulam']

(virtual_network_emsync_env) G:\Final Year PROJECT\p4_final EMSYNC\emsync-root\phase3-transfer>git add .
(virtual_network_emsync_env) G:\Final Year PROJECT\p4_final EMSYNC\emsync-root\phase3-transfer>git commit -m "Fix: Dynamic Admin View and Reset Simulation Logic"
[main 72a0ad8] Fix: Dynamic Admin View and Reset Simulation Logic
 2 files changed, 58 insertions(+), 2 deletions(-)

(virtual_network_emsync_env) G:\Final Year PROJECT\p4_final EMSYNC\emsync-root\phase3-transfer>git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 777 bytes | 155.00 KiB/s, done.
Total 6 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/Alan-JPV/EMSYNC_DOCKER_BACKEND.git
  29509cb..72a0ad8  main -> main

(virtual_network_emsync_env) G:\Final Year PROJECT\p4_final EMSYNC\emsync-root\phase3-transfer>
```