

# TMT (Triage & Monitor for Threats)

## Comprehensive Project Report

Version: 1.1.0Report Date: February 19, 2026Classification: Internal Technical DocumentationPrepared By: Development Team

### Table of Contents

1. [Executive Summary](#)

2. [System Architecture](#)

3. [Technology Stack](#)

4. [Feature Inventory](#)

5. [Backend Services](#)

6. [Frontend Application](#)

7. [AI & Machine Learning](#)

8. [Communication Infrastructure](#)

9. [Security & Compliance](#)

10. [Database Schema](#)

11. [API Reference](#)

12. [Deployment Infrastructure](#)

13. [Testing & Quality Assurance](#)

14. [Performance Metrics](#)

15. [Third-Party Integrations](#)

16. [Risk Assessment](#)

17. [Appendices](#)

## 1. Executive Summary

### 1.1 Project Overview

**TMT (Triage & Monitor for Threats)** is an enterprise-grade emergency response platform designed for crisis management in conflict zones and disaster scenarios. The system connects citizens in emergency situations with appropriate response teams through an AI-powered triage pipeline, resilient multi-channel communication, and real-time situational awareness.

### 1.2 Key Capabilities

Capability	Description
Intelligent SOS	One-tap emergency trigger with AI-assisted triage conversation
Resilient Communication	Triple-layer fallback: Internet → SMS → Bluetooth Mesh
AI Crisis Intelligence	Real-time monitoring of Telegram channels with GLM-5 classification
Multi-Department Response	Hospital, Police, Civil Defense coordination
Field Responder Interface	Mobile apps for Ambulance, Police, Civil Defense, Firefighter
Live Situational Map	Real-time geospatial visualization of incidents

### 1.3 Project Metrics

Metric	Value
--------	-------

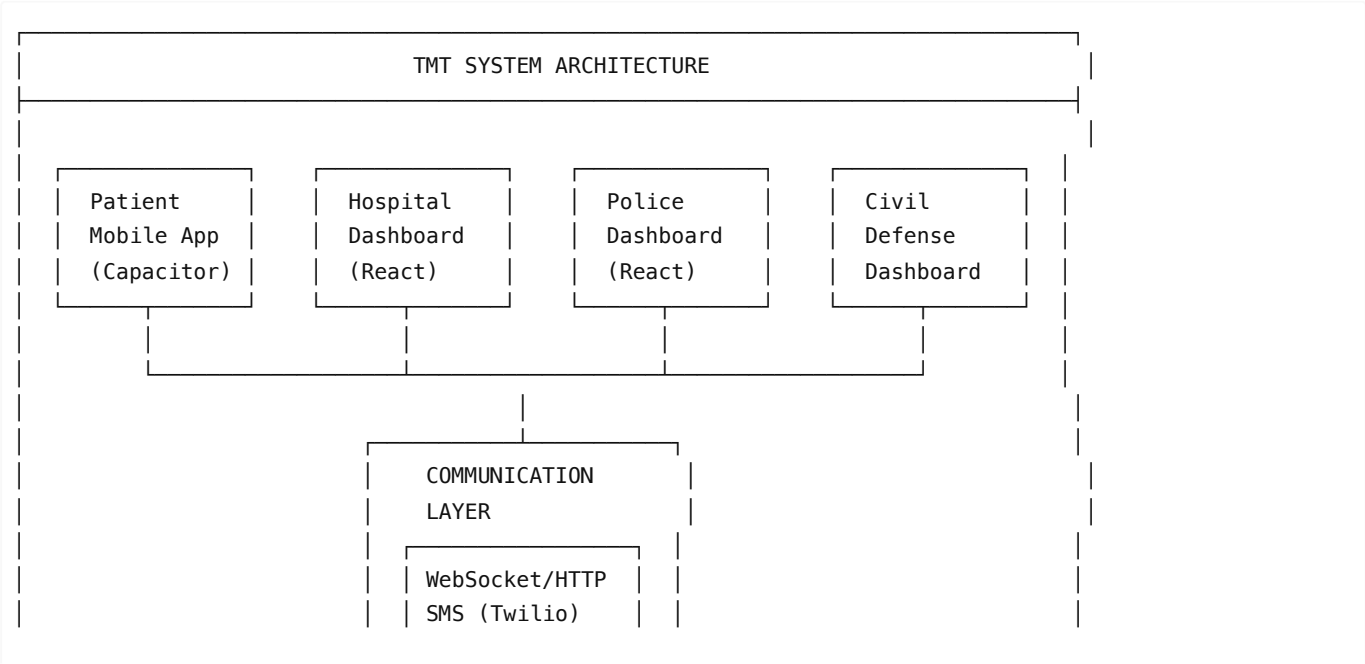
Total Source Files	256 files
Total Lines of Code	54,771 lines
Backend Modules	50+ Python modules
Frontend Components	30+ React components
API Endpoints	100+ endpoints across 16 route modules
Database Models	14 SQLAlchemy ORM models
Test Coverage	Backend: 20+ tests, Frontend: Vitest configured

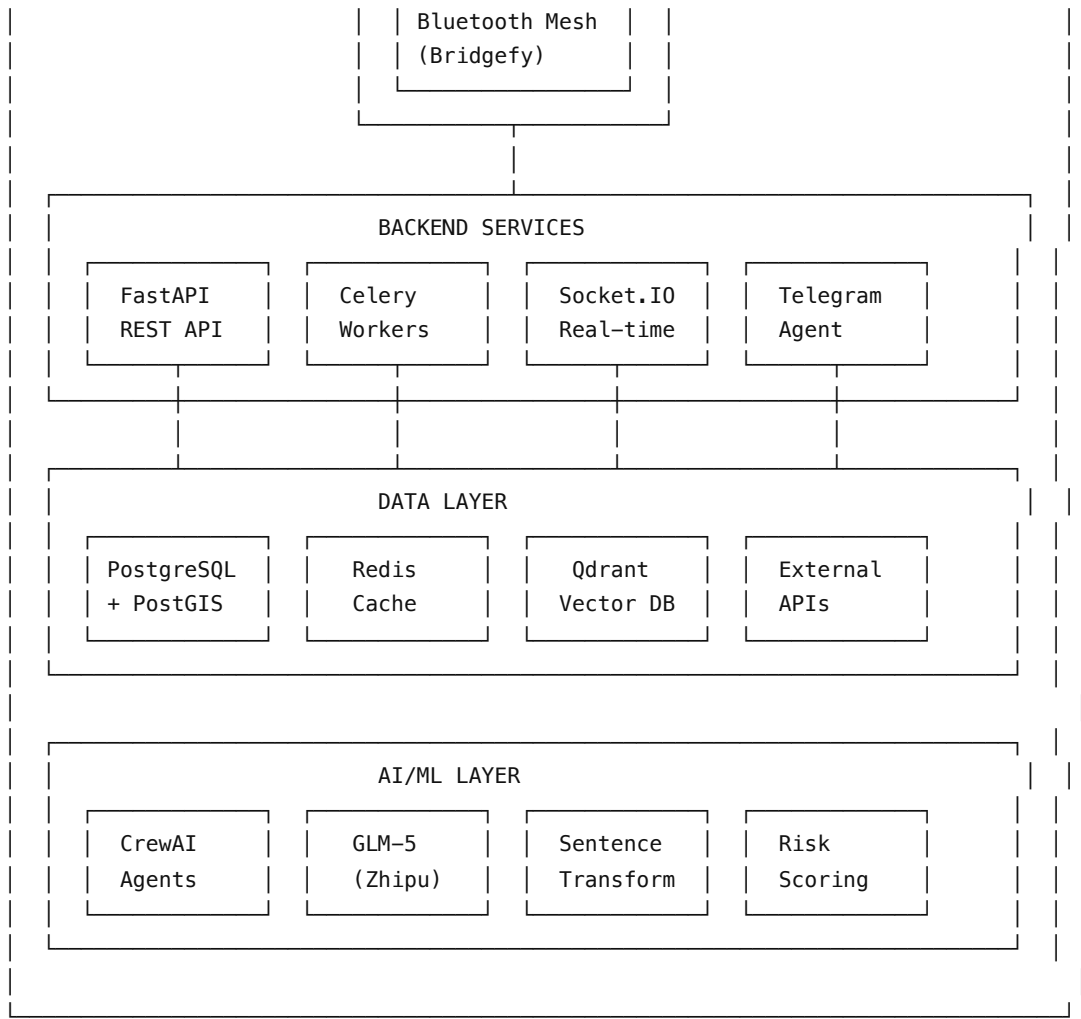
1.4 Stakeholder Matrix

Role	Platform	Access Level
Super Admin (Ministry)	Web + Mobile	Full system oversight
Hospital Admin	Web + Mobile	Patient intake, resource management
Police Admin	Web + Mobile	Security incident management
Civil Defense Admin	Web + Mobile	Rescue coordination
Ambulance Driver	Mobile	Active case navigation
Police Officer	Mobile	Field security response
Civil Defense Responder	Mobile	Field rescue operations
Firefighter	Mobile	Field fire/rescue operations
Patient/Citizen	Mobile	SOS submission, news feed

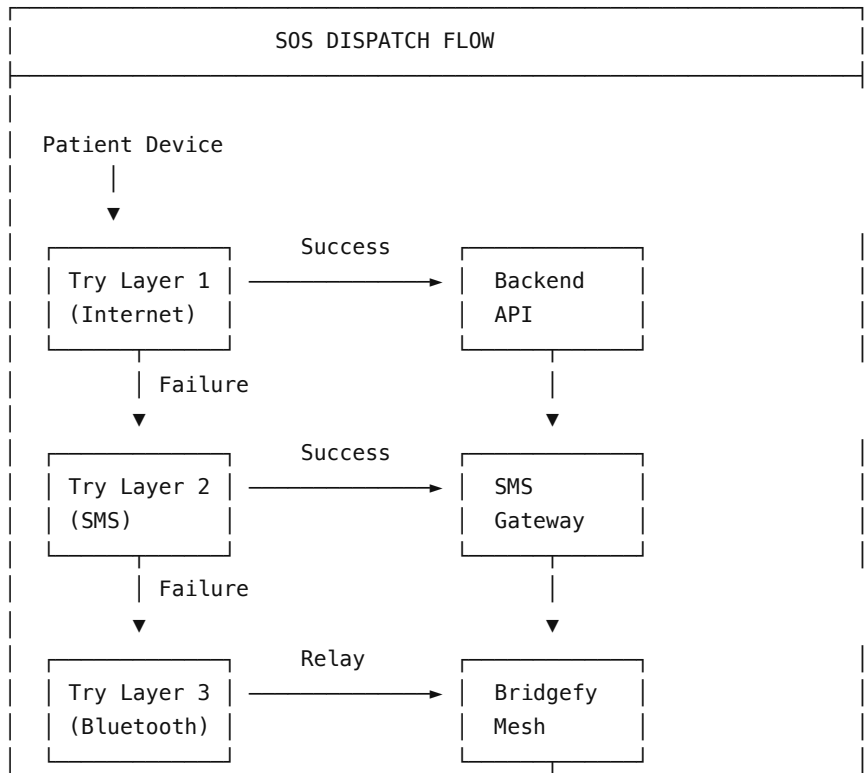
2. System Architecture

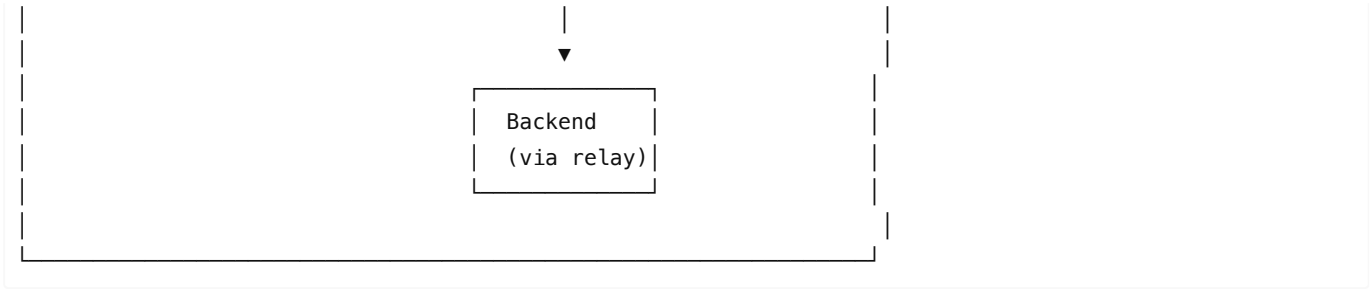
2.1 High-Level Architecture





2.2 Service Communication Flow





### 2.3 Microservices Topology

Service	Port	Technology	Purpose
PostgreSQL	5432	PostgreSQL 16 + PostGIS	Primary database with geospatial support
Redis	6379	Redis 7	Message broker, cache, rate limiting
Qdrant	6333/6334	Qdrant 1.7.4	Vector database for semantic search
Backend API	8000	FastAPI + Uvicorn	REST API + WebSocket
Celery Worker	—	Celery 5.3	Async task processing
Celery Beat	—	Celery Beat	Scheduled task coordination
Frontend	3000	React 19 + Vite	Web dashboard

## 3. Technology Stack

### 3.1 Backend Technologies

Category	Technology	Version	Purpose
Framework	FastAPI	0.109+	Async Python web framework
Database	PostgreSQL	16	Primary relational database
Geospatial	PostGIS	3.4	Geographic data extension
ORM	SQLAlchemy	2.0+	Object-relational mapping
Task Queue	Celery	5.3+	Distributed task processing
Cache/Broker	Redis	7.x	In-memory data store
Vector DB	Qdrant	1.7.4	Semantic similarity search
Real-time	Socket.IO	5.11+	WebSocket communication
Auth	python-jose	3.3.0	JWT token handling
Password	passlib + bcrypt	1.7.4	Secure password hashing
Encryption	cryptography	42.0+	Data encryption
SMS	Twilio	8.13+	SMS gateway
Telegram	Telethon	1.34	Telegram MTProto client
AI Framework	CrewAI	1.9.3	Multi-agent orchestration

LLM	LiteLLM	1.50+	LLM abstraction layer
Embeddings	sentence-transformers	2.5+	Text embeddings
Geospatial Lib	Shapely	2.0+	Geometric operations
Geocoding	Geopy	2.4+	Reverse geocoding

3.2 Frontend Technologies

Category	Technology	Version	Purpose
Framework	React	19.0.0	UI component library
Language	TypeScript	5.7	Type-safe JavaScript
Build Tool	Vite	6.1	Fast development bundler
Styling	Tailwind CSS	4.0	Utility-first CSS
State	Zustand	5.0	Minimal state management
Routing	React Router	7.1	Client-side routing
Maps	Leaflet	1.9	Interactive maps
Charts	Recharts	2.15	Data visualization
Real-time	Socket.IO Client	4.8	WebSocket client
i18n	i18next	24.2	Internationalization
Native Bridge	Capacitor	8.1	iOS/Android native APIs
Testing	Vitest	4.0	Unit testing framework

3.3 Infrastructure Technologies

Category	Technology	Version	Purpose
Containerization	Docker	Latest	Service containerization
Orchestration	Docker Compose	Latest	Multi-container management
Database Image	postgis/postgis	16-3.4	PostgreSQL with PostGIS
Cache Image	redis	7-alpine	Lightweight Redis
Vector DB Image	qdrant/qdrant	v1.7.4	Vector search engine

3.4 External Services

Service	Provider	Purpose
SMS Gateway	Twilio	Send/receive SMS
LLM API	Zhipu AI (GLM-5)	AI reasoning and classification
Social Monitoring	Telegram	Crisis intelligence gathering
Mesh Network	Bridgefy	Offline peer-to-peer communication

---

## 4. Feature Inventory

### 4.1 SOS Emergency System

Feature	Status	Description
One-Tap SOS Button	✔ Complete	Large, always-visible emergency trigger
5-Second Cancel Window	✔ Complete	Countdown before SOS transmission
GPS Auto-Capture	✔ Complete	Location captured immediately on trigger
AI Triage Conversation	✔ Complete	Post-SOS AI chat for structured data collection
Voice Input	✔ Complete	Speech-to-text with waveform animation
Quick-Tap Options	✔ Complete	Pre-defined response chips
Photo Attachment	✔ Complete	Situational photo capture
Unresponsive Auto-Send	✔ Complete	Auto-submit after 30s no response
Low Battery Mode	✔ Complete	Expedited 2-question flow when battery < 15%
Urgency Keyword Detection	✔ Complete	Immediate send on crisis keywords
Offline Queuing	✔ Complete	IndexedDB storage when offline
WebSocket Broadcast	✔ Complete	Real-time notification to stakeholders
Geofence Auto-Close	✔ Complete	SOS auto-resolved at hospital entry
Triage Transcript Storage	✔ Complete	Full conversation saved with SOS

### 4.2 Multi-Channel Communication

Layer	Status	Technology	Fallback Trigger
Internet (Primary)	✔ Active	WebSocket + HTTPS	—
SMS (Secondary)	✔ Active	Twilio Gateway	No internet
Bluetooth Mesh (Tertiary)	✔ Implemented	Bridgefy SDK	No internet + no cellular

### 4.3 Crisis Intelligence

Feature	Status	Description
Telegram Channel Monitoring	✔ Active	Automated message extraction
GLM-5 Classification	✔ Active	AI-powered crisis detection
Vector Semantic Search	✔ Active	Qdrant similarity queries
Knowledge Gap Detection	✔ Active	Coverage analysis
Trust Scoring	✔ Active	Per-source reliability score
Event Deduplication	✔ Active	Prevent duplicate alerts

### 4.4 Web Dashboard

Module	Status	Description
KPI Overview	✔ Complete	Active SOS, dispatched units, resolved cases
Real-Time SOS List	✔ Complete	Live-updating emergency list
AI Triage Score Display	✔ Complete	0-100 priority scoring
Patient Medical Summary	✔ Complete	Medical context per case
AI Recommendations	✔ Complete	Equipment/procedure suggestions
Case Assignment	✔ Complete	Unit/responder dispatch
Live Map	✔ Complete	Geospatial incident visualization
Crisis Heatmap	✔ Complete	Density visualization
Analytics Charts	✔ Complete	Trends and distributions
Resource Requests	⚠ Partial	Blood/supply/equipment requests
Case Transfers	⚠ Partial	Inter-department routing

4.5 Field Responder Mobile

Responder Type	Status	Features
Ambulance Driver	✔ Complete	Case card, navigation, status flow
Police Officer	✔ Complete	Case card, navigation, status flow
Civil Defense	✔ Complete	Case card, equipment list, navigation
Firefighter	✔ Complete	Case card, equipment list, navigation

4.6 Patient/Citizen Mobile

Feature	Status	Description
Registration	✔ Complete	Phone + OTP verification
Medical Profile	✔ Complete	Conditions, allergies, medications
Emergency Contacts	✔ Complete	Up to 3 auto-notified contacts
SOS Trigger	✔ Complete	Full emergency flow
News Feed	✔ Complete	Location-aware crisis updates
Alert Notifications	✔ Complete	Push/local alerts
Profile Management	✔ Complete	Edit personal/medical info

5. Backend Services

5.1 Service Layer Architecture





## 5.2 Core Services Detail

### 5.2.1 Patient Service

- **File:** `app/services/patient_service.py` (22KB)
- **Responsibilities:**
  - Patient CRUD operations
  - Medical profile management
  - Location-based queries (PostGIS)
  - Consent tracking (GDPR)
  - Trust score calculations
  - Risk score/level updates

### 5.2.2 Alert Service

- **File:** `app/services/alert_service.py` (18KB)
- **Responsibilities:**
  - Alert creation with severity classification
  - Spatial patient matching (affected radius)
  - Vulnerable patient identification
  - Alert acknowledgment workflow
  - Department routing

### 5.2.3 Hospital Service

- **File:** `app/services/hospital_service.py` (13KB)

- **Responsibilities:**
  - Facility CRUD operations
  - Multi-department support (Hospital, Police, Civil Defense)
  - Nearest hospital queries (PostGIS)
  - Occupancy management
  - Real-time status updates

5.2.4 SMS Service

- **File:** app/services/sms\_service.py (14KB)
- **Responsibilities:**
  - Inbound SMS SOS parsing
  - Encrypted payload handling
  - Twilio integration
  - Acknowledgment sending

5.2.5 Analytics Service

- **File:** app/services/analytics\_service.py (14KB)
- **Responsibilities:**
  - Dashboard statistics
  - Heatmap data generation
  - Timeline analysis
  - Hospital occupancy tracking

5.3 Celery Task Queues

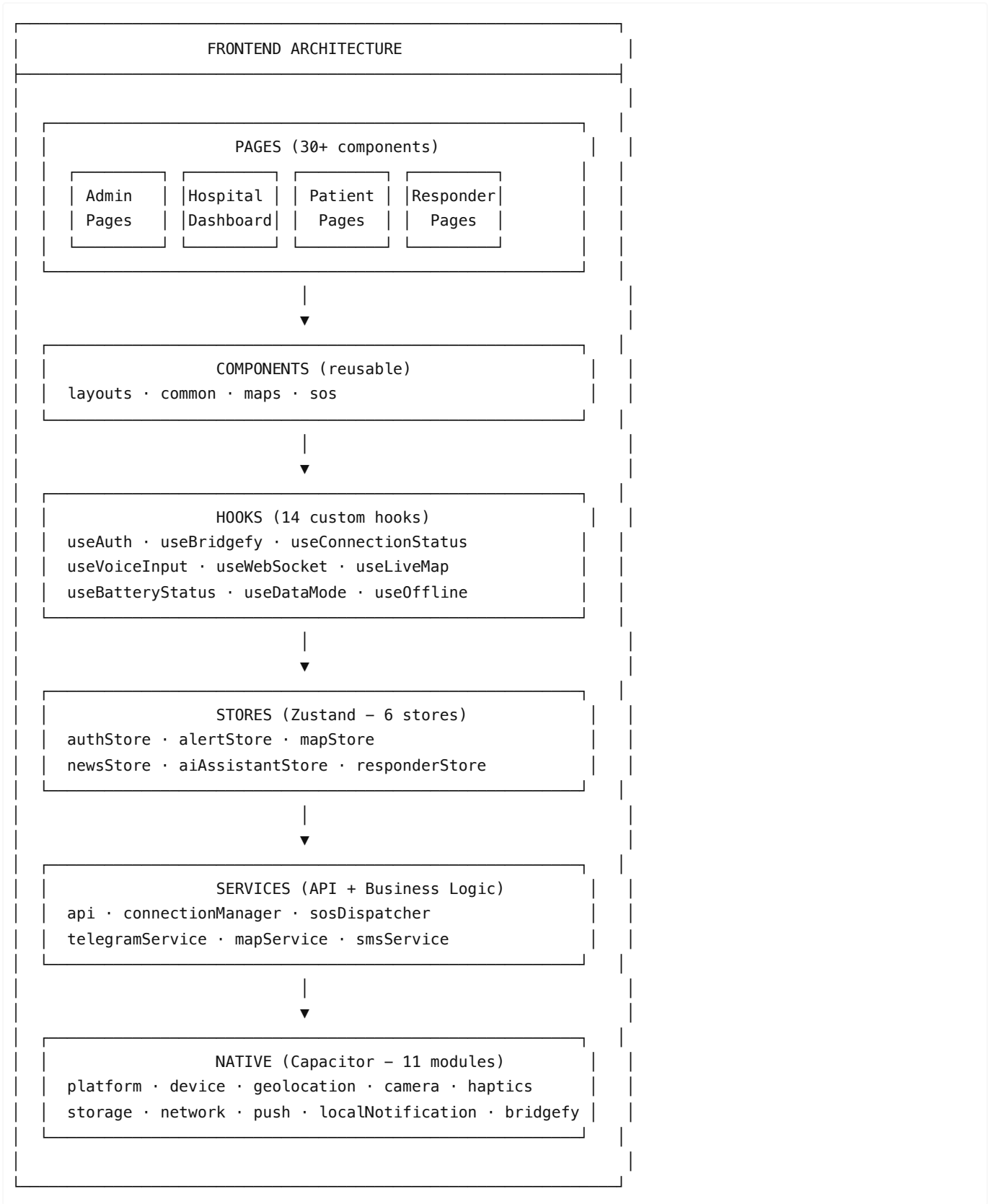
Queue	Priority	Tasks
sos.triage	HIGH	AI triage pipeline, risk scoring
alerts.new	MEDIUM	Alert creation/distribution
sms.inbound	MEDIUM	SMS processing
map.updates	MEDIUM	Geo-event streaming
intel.analysis	LOW	Telegram message analysis
embeddings.generate	BACKGROUND	Vector generation
verification	LOW	Event verification

5.4 Celery Beat Schedule

Task	Interval	Purpose
fetch_and_process_messages	5 minutes	Telegram monitoring
gap_detection_cycle	1 hour	Coverage analysis
analytics_refresh	5 minutes	Dashboard cache refresh
verify_telegram_events	30 minutes	Event validation

6. Frontend Application

6.1 Component Architecture



### 6.2 Route Configuration

Path	Component	Role Access
/login	Login	Public
/register	Register	Public

/sos	SOS	Patient
/news	News	All authenticated
/profile	Profile	Patient
/patient-alerts	PatientAlerts	Patient
/health-records	MedicalRecords	Patient
/dashboard/*	Hospital Dashboard	Hospital/Police/Civil Defense Admin
/admin/*	Admin Dashboard	Super Admin
/ambulance/*	Ambulance App	Ambulance Driver
/police/*	Police App	Police Officer
/civil_defense/*	Civil Defense App	Civil Defense Responder
/firefighter/*	Firefighter App	Firefighter

6.3 State Management (Zustand Stores)

Store	Size	State Contents
authStore	2.7KB	token, user, isAuthenticated
responderStore	14KB	currentLocation, activeCase, completedCases, isOnDuty
alertStore	2.4KB	alerts array, filter state
newsStore	6.4KB	articles, filters, unread count
aiAssistantStore	2.7KB	messages, isLoading, thinking
mapStore	1.5KB	center, zoom, selectedLocation

6.4 Native Capabilities (Capacitor)

Module	Plugin	Capability
Geolocation	@capacitor/geolocation	GPS tracking
Camera	@capacitor/camera	Photo capture
Device	@capacitor/device	Device info (UUID, model)
Network	@capacitor/network	Connectivity status
Preferences	@capacitor/preferences	Secure storage
Push Notifications	@capacitor/push-notifications	Remote notifications
Local Notifications	@capacitor/local-notifications	In-app alerts
Haptics	@capacitor/haptics	Vibration feedback
Filesystem	@capacitor/filesystem	File operations

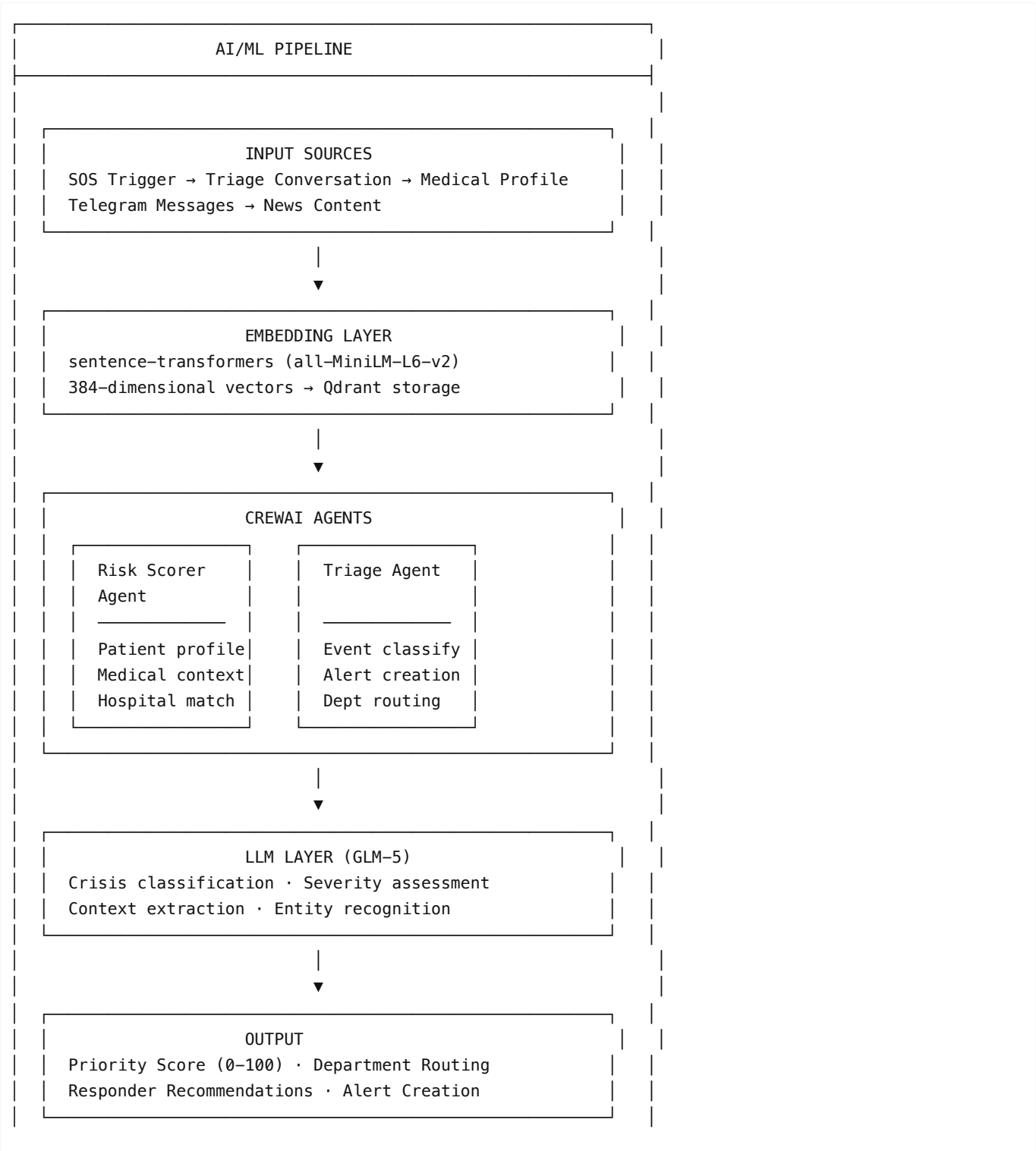
6.5 Internationalization (i18n)

Language	Code	RTL Support
Arabic	ar	✔ Yes
English	en	No

**Translation Keys:** 1000+ across all modules

## 7. AI & Machine Learning

### 7.1 AI Pipeline Architecture





7.2 CrewAI Agent Configuration

Risk Scorer Agent

- **Purpose:** Patient profile analysis and hospital recommendation
- **Inputs:** Patient medical record, triage data, location
- **Outputs:** Priority score, recommended facility

Triage Agent

- **Purpose:** Event classification and alert creation
- **Inputs:** SOS details, conversation transcript, context
- **Outputs:** Event type, severity, department routing

7.3 Vector Intelligence (Qdrant)

Configuration	Value
Collection Name	tmt_intelligence
Vector Dimension	384
Distance Metric	Cosine
Embedding Model	all-MiniLM-L6-v2

Use Cases:

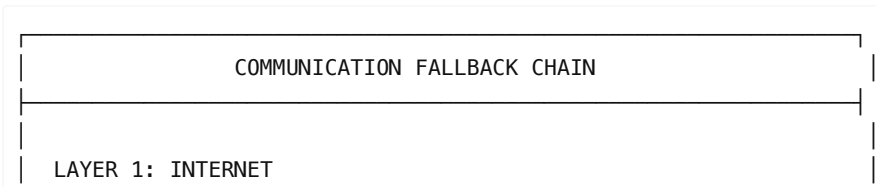
- Crisis event deduplication
- Semantic similarity search
- Knowledge gap detection
- Context enrichment for triage

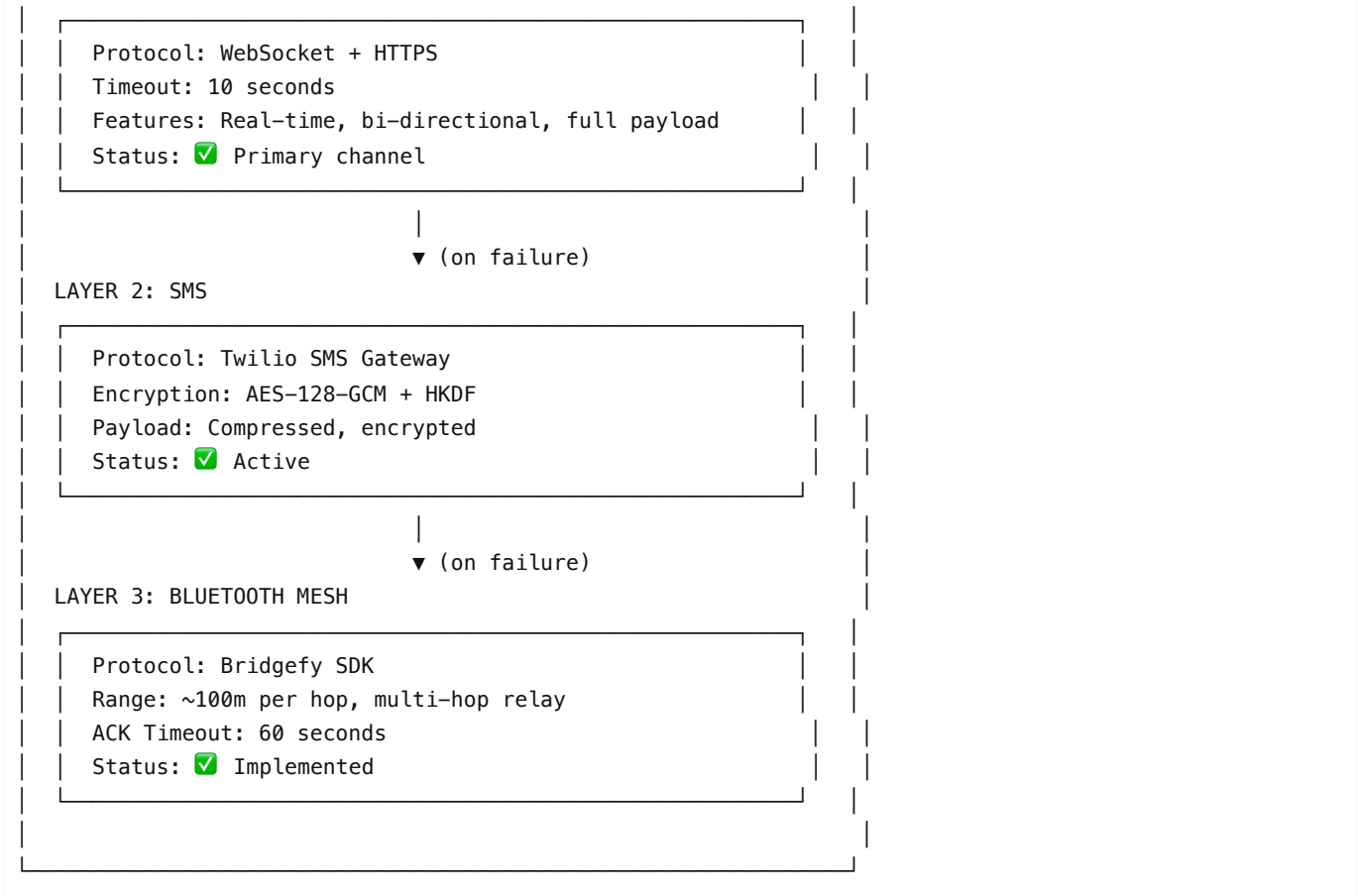
7.4 AI Scoring Factors

Factor	Weight	Description
Medical Conditions	High	Chronic conditions, allergies
Mobility Status	High	Ability to self-evacuate
Living Situation	Medium	Alone, elderly, special needs
Patient Trust Score	Medium	Historical accuracy
Location Danger Context	High	Nearby confirmed crises
Response Time	Increasing	Score increases over time

8. Communication Infrastructure

8.1 Three-Layer Fallback System





### 8.2 Connection Manager

State Property	Description
currentLayer	Active communication layer
internet.available	Internet connectivity status
internet.latency	Connection latency (ms)
internet.quality	good / poor / unknown
cellular.available	Cellular network status
cellular.signalStrength	Signal strength (0-5)
cellular.canSendSMS	SMS capability
bluetooth.meshConnected	Bridgefy mesh status
bluetooth.nearbyDevices	Peer device count

### 8.3 SOS Dispatcher

File: frontend/src/services/sosDispatcher.ts

Method	Purpose
dispatch(payload)	Send SOS through fallback chain
retryPending()	Retry queued SOS messages

getPendingCount()	Get offline queue count
-------------------	-------------------------

8.4 WebSocket Events

Room	Events
hospital_{id}	SOS, alerts, status updates
dept_{department}	Department-wide broadcasts
patient_{id}	Personal notifications
alerts	Global alert broadcasts
livemap	Geo-event updates
telegram	News feed updates

9. Security & Compliance

9.1 Authentication & Authorization

Mechanism	Implementation
Token Type	JWT (HS256)
Token Lifetime	24 hours
Password Hashing	bcrypt (passlib)
Role-Based Access	Custom decorators + middleware
Session Storage	Client-side (localStorage)

9.2 Encryption Standards

Data Type	Algorithm	Key Size
Medical Records (at rest)	AES-256-CBC	256-bit
SMS Payloads	AES-128-GCM	128-bit
Key Derivation	HKDF (SHA-256)	256-bit
Transport	TLS 1.3	256-bit

9.3 GDPR Compliance

Requirement	Status	Implementation
Explicit Consent	✓	Consent form + database tracking
Data Minimization	✓	Only emergency-relevant data
Right to Access	✓	Profile viewing
Right to Erasure	■ Planned	Account deletion endpoint
Data Portability	■ Planned	Export functionality

Breach Notification	<div><div></div>Planned</div>	Alert system
Audit Trail	<div><div></div></div>	audit_log table

9.4 API Security

Protection	Implementation
Rate Limiting	200 requests/60s (Redis sliding window)
CORS	Configured (needs production tightening)
Input Validation	Pydantic schemas
SQL Injection	SQLAlchemy ORM (parameterized)
XSS Prevention	React auto-escaping

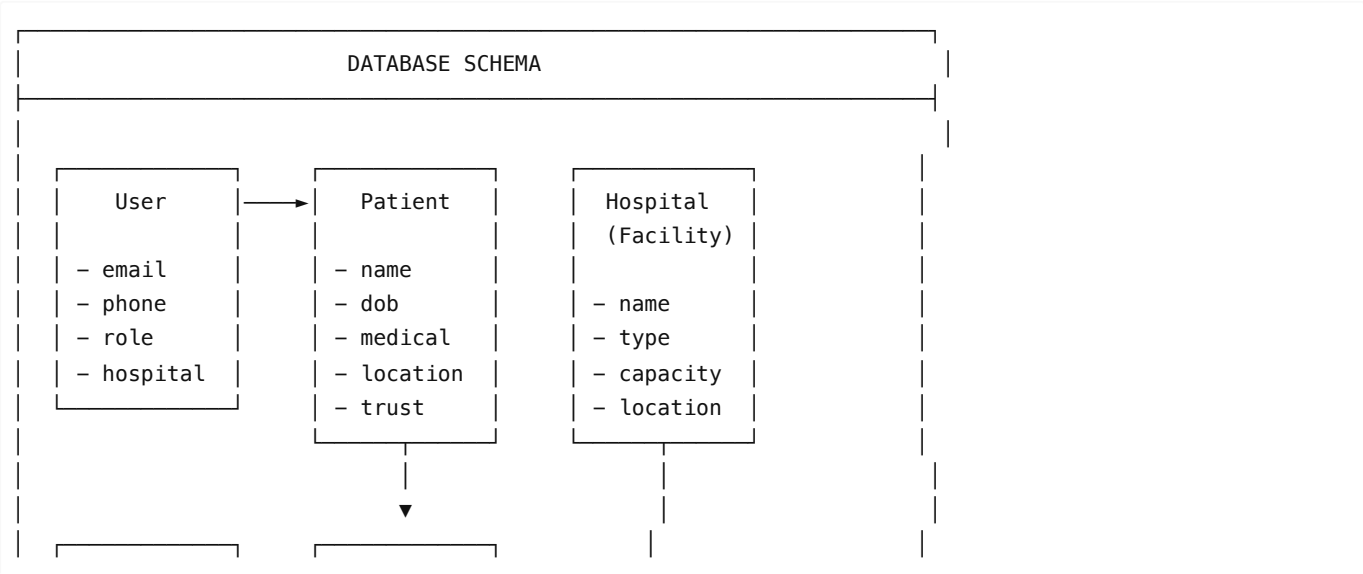
9.5 Audit Logging

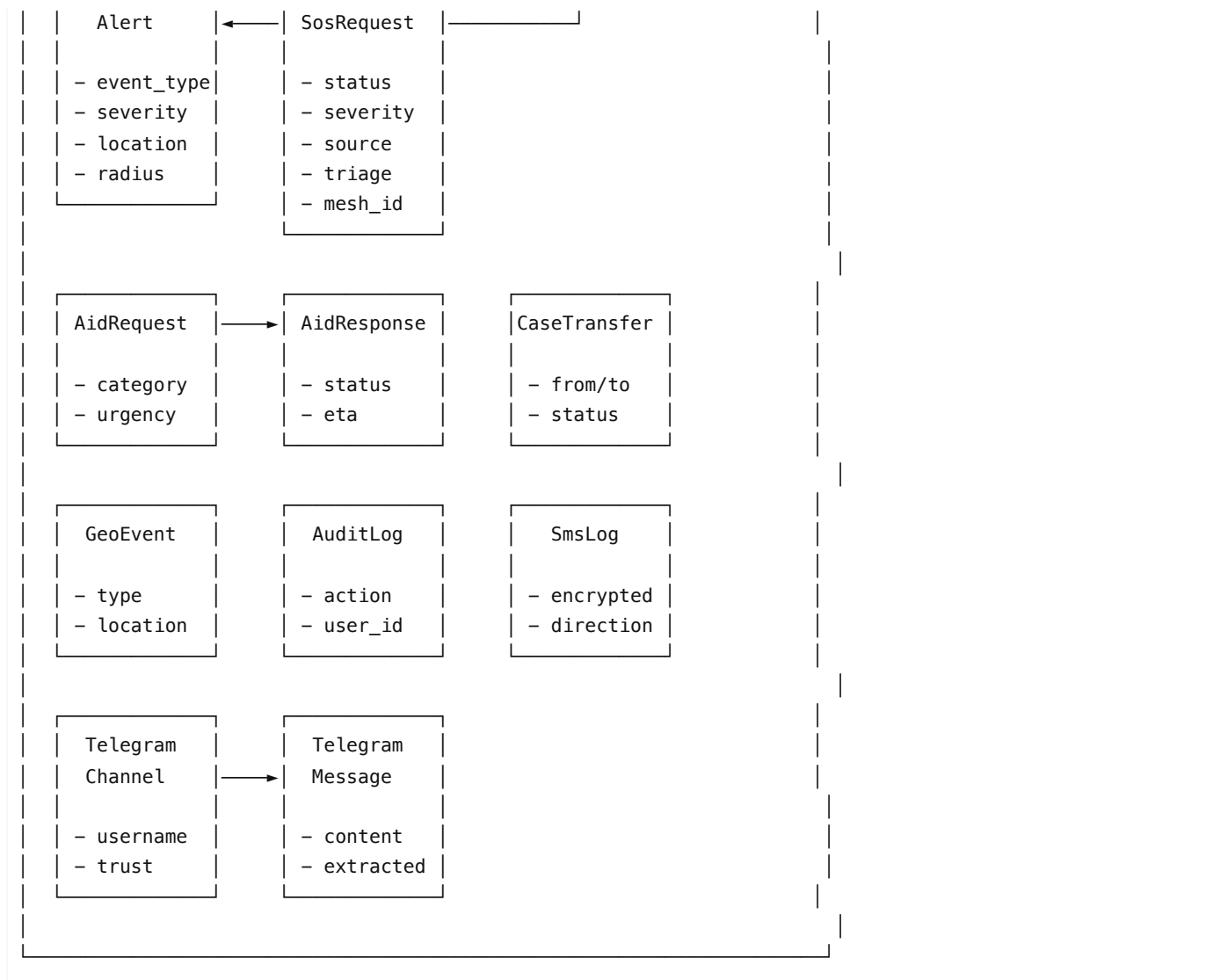
Table: audit\_log

Field	Description
user_id	Acting user
action	Operation type
resource	Affected resource type
resource_id	Affected resource ID
ip_address	Client IP
user_agent	Client user-agent
timestamp	Action timestamp

10. Database Schema

10.1 Entity-Relationship Overview





## 10.2 Core Models

Model	Key Fields	Purpose
User	email, phone, role, hospital_id	Authentication & roles
Patient	name, medical_data, trust_score, risk_level	Citizen profiles
Hospital	name, department_type, capacity, location	Facility management
SosRequest	status, severity, source, mesh_message_id	Emergency tracking
Alert	event_type, severity, location, radius_m	Crisis events
AidRequest	category, urgency, status	Resource requests
AidResponse	status, eta_hours	Request responses
CaseTransfer	from_facility, to_facility, status	Inter-dept routing
AuditLog	action, user_id, resource, timestamp	Compliance trail

## 10.3 Enums

Enum	Values
------	--------

UserRole	patient, hospital_admin, police_admin, civil_defense_admin, super_admin
DepartmentType	HOSPITAL, POLICE, CIVIL_DEFENSE
SOSStatus	PENDING, ACKNOWLEDGED, DISPATCHED, RESOLVED, CANCELLED
SOSSource	API, SMS, MESH
PatientStatus	SAFE, INJURED, TRAPPED, EVACUATE
AlertSeverity	LOW, MEDIUM, HIGH, CRITICAL
EventType	BOMBING, SHOOTING, CHEMICAL, FIRE, BUILDING_COLLAPSE, EARTHQUAKE, FLOOD, etc.

## 11. API Reference

### 11.1 Endpoint Summary

Module	Base Path	Endpoints
Auth	/auth	login
Patients	/patients	CRUD, location queries
Hospitals	/hospitals	CRUD, nearest queries
SOS	/sos	create, list, update status
Alerts	/alerts	create, list, acknowledge
Mesh	/mesh	relay, ack, heartbeat, stats
Telegram	/telegram	channels, messages, status
SMS	/sms	inbound processing
Live Map	/livemap	geo-events, heatmap
Analytics	/analytics	dashboard, statistics
Aid Requests	/aid_requests	CRUD, responses
Transfers	/transfers	create, accept, reject

### 11.2 Key Endpoints

#### Authentication

```
POST /auth/login
Body: { phone, password }
Response: { access_token, user }
```

#### SOS

```
POST /sos
Body: { latitude, longitude, patient_status, severity, details, triage_transcript }
Response: { id, status, created_at }

GET /sos
Response: [{ id, patient, status, severity, created_at }]
```

```
PATCH /sos/{id}/status
Body: { status }
```

Mesh Relay

```
POST /mesh/relay
Body: { message_id, patient_id, latitude, longitude, patient_status, severity, hop_count,
relay_device_id }
Response: { sos_id, is_duplicate }
```

Alerts

```
POST /alerts
Body: { title, event_type, severity, latitude, longitude, radius_m }
Response: { id, affected_patients_count }

GET /alerts
Response: [{ id, title, severity, location, created_at }]
```

12. Deployment Infrastructure

12.1 Docker Compose Services

```
services:
  postgres:      # PostgreSQL 16 + PostGIS
  redis:         # Redis 7 Alpine
  qdrant:        # Qdrant v1.7.4
  backend:       # FastAPI application
  celery-worker: # Celery workers (4 concurrent)
  celery-beat:   # Scheduled tasks
  frontend:     # React + Vite dev server
```

12.2 Environment Variables

Variable	Required	Description
POSTGRES_PASSWORD	Yes	Database password
JWT_SECRET	Yes	JWT signing key (32+ bytes)
ENCRYPTION_MASTER_KEY	Yes	AES master key (32 bytes)
TWILIO_ACCOUNT_SID	No	Twilio account
TWILIO_AUTH_TOKEN	No	Twilio auth
TWILIO_PHONE_NUMBER	No	Twilio phone
GLM_API_KEY	No	Zhipu AI API key
TELEGRAM_API_ID	No	Telegram app ID
TELEGRAM_API_HASH	No	Telegram app hash
TELEGRAM_PHONE	No	Telegram phone

12.3 Volume Mounts

Volume	Path	Purpose
postgres_data	/var/lib/postgresql/data	Database persistence
redis_data	/data	Redis snapshots
qdrant_data	/qdrant/storage	Vector storage

12.4 Healthchecks

Service	Check	Interval
PostgreSQL	pg_isready	5s
Redis	redis-cli ping	5s
Qdrant	TCP port 6333	5s

12.5 Production Recommendations

Area	Recommendation
Reverse Proxy	Nginx with SSL termination (Let's Encrypt)
Load Balancing	HAProxy or AWS ALB
Secrets	AWS Secrets Manager or HashiCorp Vault
Monitoring	Prometheus + Grafana
Logging	ELK Stack (Elasticsearch, Logstash, Kibana)
Backup	Automated PostgreSQL backups to S3
CI/CD	GitHub Actions or GitLab CI
WAF	Cloudflare or AWS WAF

13. Testing & Quality Assurance

13.1 Backend Testing

Test Suite	Framework	Count
Mesh Routes	pytest	20 tests
Department Routing	pytest	—
Fallback Chain	pytest	—
AI Agent Pipeline	pytest	—

13.2 Frontend Testing

Category	Framework	Status
Unit Tests	Vitest	Configured

Component Tests	@testing-library/react	Configured
Coverage	@vitest/coverage-v8	Configured

13.3 Test Commands

```
# Backend
cd backend
pytest tests/

# Frontend
cd frontend
npm test           # Watch mode
npm test:run       # Single run
npm test:coverage  # With coverage
```

13.4 Quality Tools

Tool	Purpose
ESLint	JavaScript/TypeScript linting
TypeScript	Static type checking
Prettier	Code formatting
Vitest	Frontend testing
pytest	Backend testing

14. Performance Metrics

14.1 Target SLAs

Metric	Target
SOS Submission	< 2s response
API Response (P95)	< 500ms
WebSocket Latency	< 100ms
Map Load Time	< 3s

14.2 Scalability

Component	Scaling Strategy
Backend API	Horizontal (multiple Uvicorn workers)
Celery Workers	Horizontal (add worker containers)
PostgreSQL	Vertical (read replicas for analytics)
Redis	Cluster mode for high availability
Qdrant	Sharding for large vector collections

14.3 Rate Limits

Endpoint Category	Limit
Global	200 req/60s
Auth	10 req/60s
SOS	10 req/60s
File Upload	5 req/60s

15. Third-Party Integrations

15.1 External Services

Service	Provider	Purpose	Status
SMS Gateway	Twilio	Send/receive SMS	✔ Active
LLM API	Zhipu AI (GLM-5)	AI reasoning	✔ Active
Telegram	Telegram MTProto	Crisis monitoring	✔ Active
Mesh Network	Bridgefy	Offline communication	✔ Implemented

15.2 API Dependencies

Package	Version	Purpose
twilio	8.13+	SMS integration
telethon	1.34	Telegram client
crewai	1.9.3	Multi-agent framework
litellm	1.50+	LLM abstraction
qdrant-client	1.7+	Vector database client

15.3 Licensing Considerations

Integration	License Type	Notes
Bridgefy SDK	Commercial	Free for development, paid for production
Twilio	Pay-per-use	SMS charges apply
GLM-5	API subscription	Token-based pricing
Telegram	Free	Subject to API rate limits

16. Risk Assessment

16.1 Technical Risks

Risk	Severity	Mitigation
Single point of failure (backend)	High	Add load balancer, multiple instances

Database data loss	Critical	Automated backups, replication
API key exposure	High	Secrets management, env vars
DDoS attack	Medium	WAF, rate limiting
Third-party service outage	Medium	Fallback mechanisms, caching

16.2 Security Risks

Risk	Severity	Mitigation
JWT token theft	High	Short expiry, secure storage
SQL injection	Low	SQLAlchemy ORM (parameterized)
XSS attacks	Low	React auto-escaping
Data breach	Critical	Encryption at rest, audit logging
Unauthorized access	High	RBAC, route guards

16.3 Operational Risks

Risk	Severity	Mitigation
Network outage	High	Multi-channel fallback (SMS, Bluetooth)
Power failure	Medium	Mobile offline queuing
Staff unavailability	Low	Comprehensive documentation
Vendor lock-in	Medium	Abstraction layers (LiteLLM)

17. Appendices

17.1 File Structure

```
TMT/
├─ backend/
│  └─ app/
│     ├── main.py
│     ├── config.py
│     ├── models/ (14 files)
│     ├── api/
│     │  ├── routes/ (16 files)
│     │  ├── middleware/ (4 files)
│     │  └─ websocket/
│     ├── services/ (10+ files)
│     ├── db/
│     └─ telegram/
├─ tasks/ (8 files)
├─ tests/
├─ Dockerfile
└─ requirements.txt
├─ frontend/
└─ src/
```

```
| | | └─ App.tsx
| | | └─ pages/ (30+ files)
| | | └─ components/
| | | └─ store/ (6 files)
| | | └─ services/
| | | └─ hooks/ (14 files)
| | | └─ native/ (11 files)
| | | └─ types/
| | └─ android/
| | └─ ios/
| | └─ Dockerfile
| | └─ package.json
└─ docs/ (13 files)
└─ docker-compose.yml
```

17.2 Development Team

Name	Role	Focus Areas
Wael	AI & Fullstack	AI pipeline, security, backend
Tareq	Mobile / DevOps	React Native, CI/CD, native integrations
Mahmoud	AI & Fullstack	Telegram agent, news system, APIs
Leen	QA / Frontend	Web dashboard, UI/UX
Roa	QA / Frontend	Patient app, testing

17.3 Documentation Index

Document	Purpose
FEATURES.md	Complete feature specification
BRIDGEFY_INTEGRATION_PLAN.md	Mesh network implementation
AI_SOS_ASSISTANT_PLAN.md	AI triage system design
SOS_FALLBACK_SYSTEM_ANALYSIS.md	Communication chain analysis
MOBILE_BACKEND_INTEGRATION_PLAN.md	Mobile app integration
PWA_TO_NATIVE_CONVERSION_PLAN.md	Native app migration
IMPLEMENTATION_TASKS.md	Development task tracking
INTEGRATION_STATUS_REPORT.md	Current progress

17.4 Glossary

Term	Definition
SOS	Emergency distress signal triggered by citizen
Triage	AI-assisted priority assessment process
Mesh Network	Peer-to-peer Bluetooth communication (Bridgefy)
PostGIS	Geographic Information System extension for PostgreSQL

Qdrant	Vector database for semantic similarity search
CrewAI	Multi-agent AI orchestration framework
GLM-5	Large Language Model from Zhipu AI
Geofence	Virtual geographic boundary for automatic actions
Trust Score	Reliability metric for patients and sources

**Document Version:** 1.0.0 **Last Updated:** February 19, 2026 **Classification:** Internal Technical Documentation

*TMT (Triage & Monitor for Threats) - Emergency Response Platform*