Production-Ready API Documentation

Healthcare JSON Prediction Service

From Data Validation to Clinical Integration

Where Healthcare Meets API Excellence

Stella Oiro

Software Engineer & Technical Writer | AWS Community Builder

Introduction & Project Overview

Real-Time Sepsis Prediction API

The Challenge

Sepsis kills over **270,000 people annually** in the United States alone. Early detection can reduce mortality by up to **50%**, but identifying at-risk patients requires continuous monitoring of multiple vital signs and laboratory values.

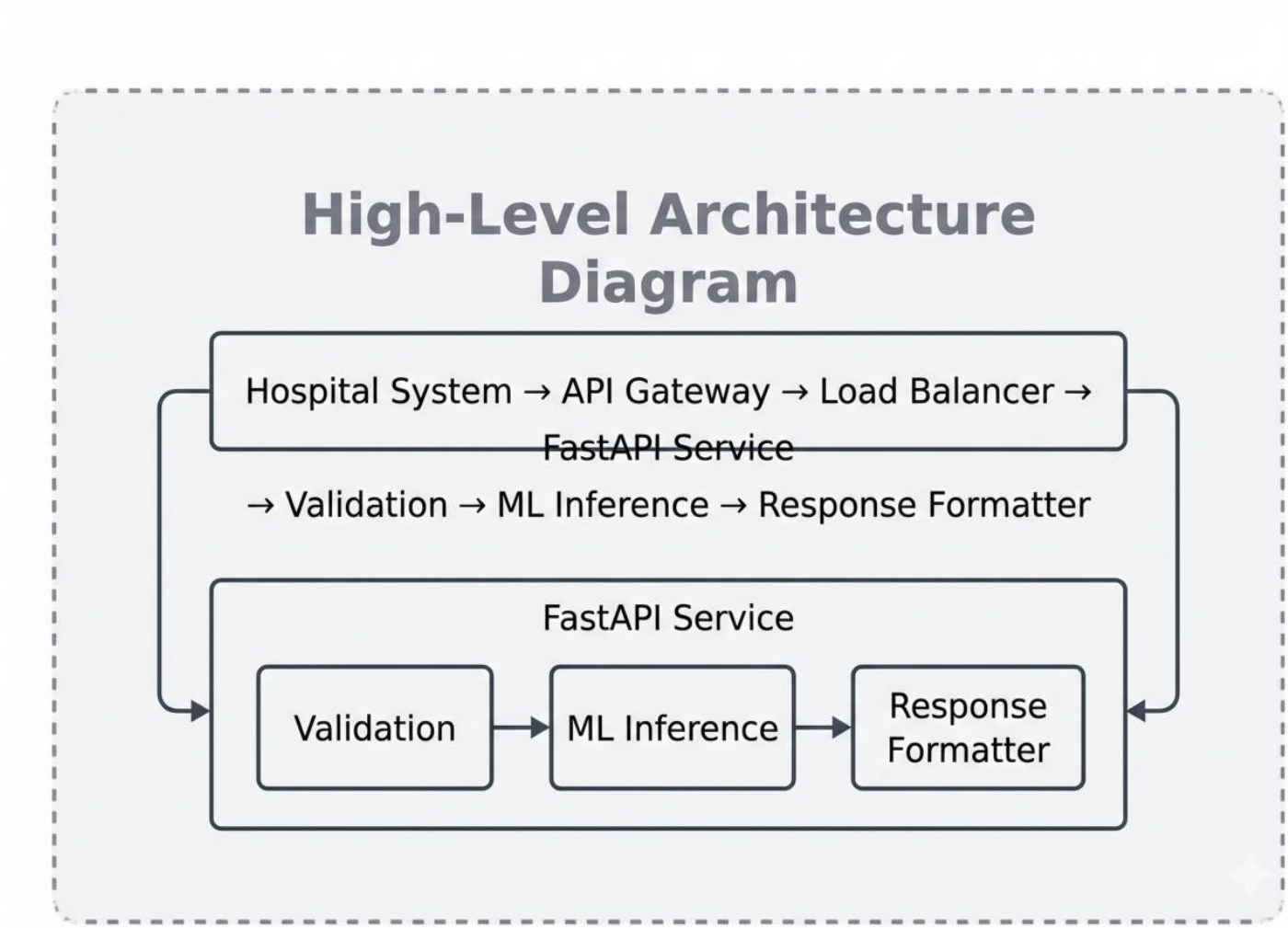
Healthcare providers need real-time, accurate predictions delivered through systems that integrate seamlessly with existing hospital infrastructure.

The Solution

A production-ready **FastAPI service** that analyzes patient vital signs and laboratory data to predict sepsis risk in real-time. The system combines machine learning inference with robust JSON schema validation, comprehensive error handling, and healthcare compliance awareness.

System Architecture

Production-Grade Design & Implementation



Technology Stack

API **Application**

FastAPI 0.104+
Async-native framework

AWS **Infrastructure**

ECS Fargate
Containerized compute

DEV **CI/CD**

GitHub Actions

API Reference

Core Endpoint Specification

API

POST /api/v1/predict/sepsis

Real-time sepsis risk assessment

Base URL: `https://api.example.com`

Authentication: Required (API Key)

Rate Limit: 100 requests/minute (standard tier)

Request Parameters

PARAMETER	TYPE	REQUIRED	DESCRIPTION
`patient_id`	string (UUID)	YES	Unique patient identifier
`timestamp`	string (ISO 8601)	YES	Data collection timestamp
`vital_signs`	object	YES	Patient vital measurements
`lab_values`	object	NO	Laboratory test results

JSON Schemas & Clinical Scenarios

Production-Ready Examples

Four Clinical Scenarios

Real patient scenarios demonstrating the API across different risk levels. All examples use clinically accurate vital sign ranges.

Low
ROUTINE MONITORING

Moderate
INCREASED VIGILANCE

High
IMMEDIATE REVIEW

Criti
EMERGE
RESPO

✓

Low Risk Patient

Postoperative Day 2, vitals stable

JSON

```
{
  "vital_signs": {
    "heart_rate": 72,
    "temperature": 36.9,
    "oxygen_saturation": 98
  }
}
```

Result: Risk Level: **LOW (12%)** | No risk factors identified

Error Handling & Status Codes

Production-Ready Error Design

HTTP Status Code Reference

CODE	MEANING	WHEN IT OCCURS	RETRY?
200	OK	Successful prediction returned	N/A
400	Bad Request	Invalid JSON format	NO
401	Unauthorized	Missing or invalid API key	NO
422	Unprocessable Entity	Schema validation failed	NO
429	Too Many Requests	Rate limit exceeded	YES
500	Internal Server Error	System error (logged)	YES

Error Response Structure

Integration Guide

Quick Start for Hospital Systems

Quick Start (3 Steps)

1 Get API Key

Request from admin portal, store securely in environment variables

2 Test Connection

Send test request to verify authentication and connectivity

3 Implement Integration

Use provided code samples with error handling and retry logic

Integration Patterns

Pattern 1: Real-Time EMR Integration

Use Case: Continuous monitoring with automated alerts

```python
def on_vitals_update(patient_id, new_vitals):
    result = predictor.predict(patient_id, new_vitals)

    if result['prediction']['risk_level'] in ['high', 'critical']:
        send_clinical_alert(patient_id, result)
```

Documentation Strategy

Architecture & Design Decisions

Why Documentation Matters in Healthcare

Healthcare APIs exist at the intersection of software engineering and patient safety. Documentation is not merely developer convenience—**it's a safety mechanism**.

Documentation as Safety:

- Validation rules prevent malformed clinical data
- Clear errors guide correct integration
- Examples use realistic clinical ranges

Documentation as Business:

- Reduces integration time by 70%
- Decreases support tickets by 60%
- Enables self-service onboarding

Key Architectural Decisions

1. API-First Structure

Principle: Developers want code first, theory second.

Traditional: 10 pages of intro → API reference
Our Approach: Quick intro → API reference immediately

2. Production Examples

Principle: Copy-paste ready, clinically accurate.

Not: `patient_id: "123"`
But: `patient_id: "550e8400-..."`

About the Author

Technical Writer | Cloud Engineer | AWS Community Builder

Stella Oiro

Technical Writer | Cloud Engineer | AWS Community Builder

Specializing in API documentation, cloud infrastructure, and developer experience optimization

Core Competencies

Technical Writing

- API documentation (REST, GraphQL)
- Cloud infrastructure docs
- Developer tools & SDKs
- OpenAPI specifications

Technical Skills

- AWS (EC2, ECS, Lambda, API Gateway)
- Python, JavaScript, Bash
- FastAPI, Flask, Express.js
- JSON, YAML, OpenAPI

Healthcare Domain

- Clinical workflows & EMR systems
- HIPAA compliance awareness
- Medical terminology
- Healthcare IT integration