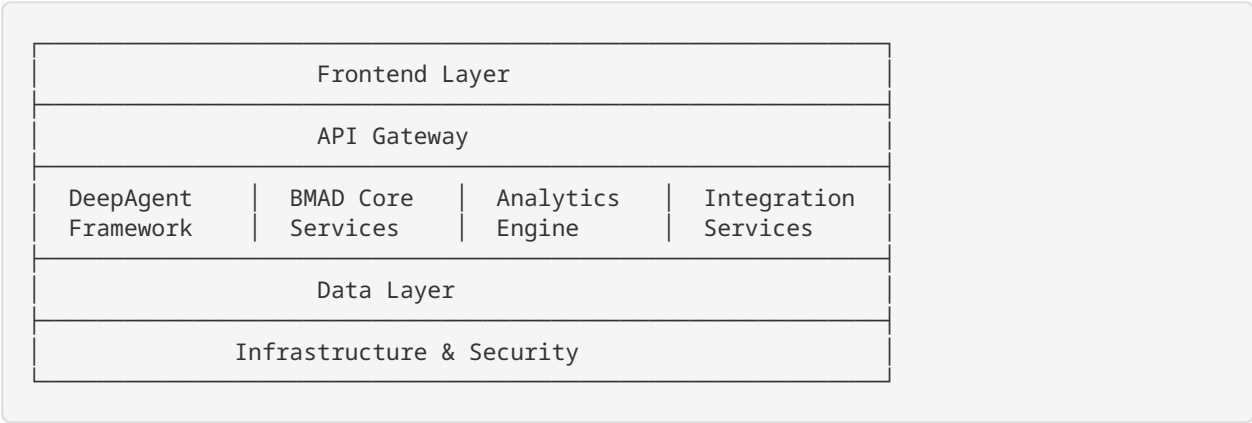


# Architecture Analysis - BioSpark Health AI System

## System Architecture Overview

### High-Level Architecture

The BioSpark Health AI system follows a modern microservices architecture with the following key components:



## Core Components

### 1. Frontend Layer

**Technology Stack:** Next.js 14, TypeScript, Tailwind CSS

- **User Interface:** Responsive web application
- **Real-time Updates:** WebSocket connections for live data
- **State Management:** Zustand for client-side state
- **Authentication:** JWT-based session management

**Key Features:**

- Dashboard for health analytics
- Agent interaction interface
- Real-time monitoring displays
- Mobile-responsive design

### 2. API Gateway

**Technology:** Kong Gateway with custom plugins

- **Rate Limiting:** 1000 requests/minute per user
- **Authentication:** JWT validation and refresh
- **Load Balancing:** Round-robin with health checks
- **Monitoring:** Request/response logging and metrics

**Endpoints Structure:**

```

/api/v1/
├── auth/           # Authentication services
├── agents/         # DeepAgent management
├── bmad/           # BMAD core functionality
├── analytics/      # Data analytics endpoints
├── health/         # System health monitoring
└── integration/    # External system integration

```

### 3. DeepAgent Framework

**Technology:** Python 3.11, FastAPI, Celery

- **Agent Orchestration:** Multi-agent coordination system
- **Task Queue:** Redis-backed Celery for async processing
- **Model Management:** MLflow for model versioning
- **Communication:** gRPC for inter-agent communication

**Agent Types:**

- **Analyst Agent:** Data analysis and insights
- **Architect Agent:** System design and optimization
- **Developer Agent:** Code generation and maintenance
- **Orchestrator Agent:** Workflow coordination

### 4. BMAD Core Services

**Technology:** Node.js, Express.js, TypeScript

- **Business Logic:** Core health AI functionality
- **Data Processing:** Stream processing with Apache Kafka
- **Caching:** Redis for high-performance data access
- **Validation:** Joi for input validation

**Service Modules:**

- **Patient Management:** Health record management
- **Diagnostic Engine:** AI-powered health analysis
- **Treatment Planning:** Personalized care recommendations
- **Monitoring System:** Real-time health tracking

### 5. Analytics Engine

**Technology:** Python, Apache Spark, TensorFlow

- **Data Pipeline:** ETL processes for health data
- **Machine Learning:** TensorFlow models for predictions
- **Real-time Analytics:** Apache Kafka Streams
- **Visualization:** Plotly for interactive charts

**Analytics Capabilities:**

- **Predictive Modeling:** Health outcome predictions
- **Pattern Recognition:** Anomaly detection in health data
- **Trend Analysis:** Long-term health trend identification
- **Risk Assessment:** Personalized risk scoring

### 6. Data Layer

**Primary Database:** PostgreSQL 15 with extensions

- **Time-series Data:** TimescaleDB for health metrics
- **Document Storage:** MongoDB for unstructured data

- **Cache Layer:** Redis Cluster for high-performance access
- **Search Engine:** Elasticsearch for full-text search

#### Data Architecture:

```
-- Core Tables
patients (id, demographics, created_at, updated_at)
health_records (id, patient_id, data, timestamp)
agents (id, type, config, status)
analytics_results (id, patient_id, analysis_type, results)
```

## 7. Infrastructure & Security

**Container Orchestration:** Kubernetes with Helm charts

- **Service Mesh:** Istio for service-to-service communication
- **Monitoring:** Prometheus + Grafana for metrics
- **Logging:** ELK Stack (Elasticsearch, Logstash, Kibana)
- **Security:** Vault for secrets management

#### Security Features:

- **Encryption:** AES-256 for data at rest, TLS 1.3 for transit
- **Authentication:** OAuth 2.0 with PKCE
- **Authorization:** RBAC with fine-grained permissions
- **Compliance:** HIPAA, GDPR compliance built-in

## Integration Points

---

### External Systems

1. **EHR Systems:** HL7 FHIR integration
2. **Medical Devices:** IoT device data ingestion
3. **Laboratory Systems:** Lab result integration
4. **Pharmacy Systems:** Medication management
5. **Insurance Systems:** Claims and coverage verification

## API Specifications

```
# OpenAPI 3.0 specification
openapi: 3.0.0
info:
  title: BioSpark Health AI API
  version: 1.0.0
paths:
  /api/v1/patients:
    get:
      summary: List patients
      parameters:
        - name: limit
          in: query
          schema:
            type: integer
            default: 50
      responses:
        200:
          description: Patient list
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/Patient'
```

## Performance Specifications

### Response Time Requirements

- **API Endpoints:** < 200ms average response time
- **Database Queries:** < 50ms for simple queries
- **Complex Analytics:** < 5 seconds for real-time analysis
- **Batch Processing:** < 30 minutes for daily reports

### Scalability Targets

- **Concurrent Users:** 10,000 simultaneous users
- **API Throughput:** 50,000 requests per minute
- **Data Volume:** 1TB of health data per month
- **Agent Processing:** 1,000 concurrent agent tasks

### Availability Requirements

- **System Uptime:** 99.9% availability (8.76 hours downtime/year)
- **Disaster Recovery:** RTO < 4 hours, RPO < 1 hour
- **Backup Strategy:** Daily automated backups with 30-day retention
- **Monitoring:** 24/7 automated monitoring with alerting

## Technology Stack Summary

### Backend Technologies

- **Languages:** Python 3.11, Node.js 18, TypeScript 5.0
- **Frameworks:** FastAPI, Express.js, Next.js 14

- **Databases:** PostgreSQL 15, MongoDB 6.0, Redis 7.0
- **Message Queue:** Apache Kafka, Redis (Celery)
- **Search:** Elasticsearch 8.0

## Frontend Technologies

- **Framework:** Next.js 14 with App Router
- **Styling:** Tailwind CSS 3.3
- **State Management:** Zustand
- **Charts:** Plotly.js, Chart.js
- **Testing:** Jest, Cypress

## Infrastructure

- **Containerization:** Docker, Kubernetes
- **Cloud Platform:** AWS/Azure/GCP compatible
- **CI/CD:** GitHub Actions, ArgoCD
- **Monitoring:** Prometheus, Grafana, Jaeger
- **Security:** HashiCorp Vault, cert-manager

---

This architecture analysis provides the foundation for implementing a scalable, secure, and high-performance health AI system.