**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 15 Apr. 2025 |
| Team ID | SWTID1743610409 |
| Project Name | Tune Trails |
| Maximum Marks | 4 Marks |

**Solution Architecture:**

Tune Trail employs a cloud-native microservices architecture built on AWS to deliver scalable, high-performance music streaming with AI-powered personalization.

**Key Architecture Goals**

1. **High Availability**: 99.95% uptime SLA
2. **Scalability**: Handle 50K+ concurrent streams
3. **Personalization**: Real-time recommendation engine
4. **Cost Efficiency**: Dynamic resource allocation

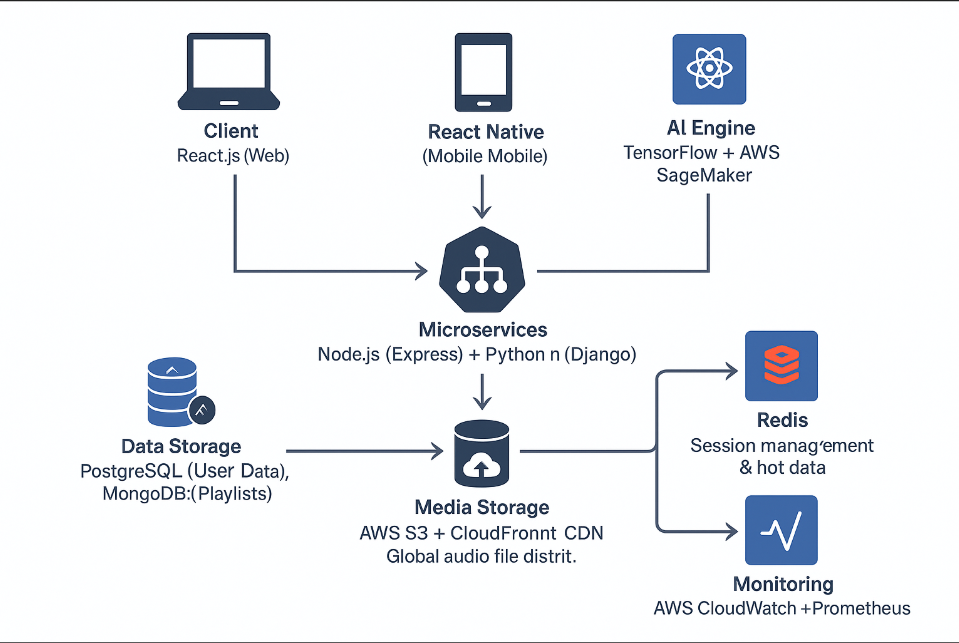
**Architecture Diagram**

|  |  |  |
| --- | --- | --- |
| **Layer** | **Technology Stack** | **Function** |
| Client | React.js (Web) | Cross-platform user interfaces |
| API Gateway | AWS API Gateway + Lambda | Request routing & authentication |
| Microservices | Node.js (Express) | Playlist generation, user profiles, analytics |
| AI Engine | TensorFlow + AWS SageMaker | Mood-based recommendations |
| Data Storage | MongoDB (Playlists) | Structured data management |
| Media Storage | AWS S3 | Global audio file distribution |
| Cache | Redis | Session management & hot data |
| Monitoring | AWS CloudWatch | Performance tracking |

Data Flow

1. User Request → API Gateway → Authentication
2. Stream Request → Media Service → S3
3. Interaction Data → Analytics Service → Recommendation Engine
4. Playlist Updates → Sync Service → All Devices

**Example - Solution Architecture Diagram:**

****

*Figure 1: Architecture and data flow of the voice patient diary sample application*