**System Architecture Blueprint — Services and Integration Layers**

**Objective:**

Establish a robust, modular, and future-proof architecture to support secure, scalable, and interoperable HR services on mobile and web platforms, especially in resource-constrained or remote field environments.

**Components:**

**A. Architecture Style:**

* **Service-Oriented Architecture (SOA)** with loosely coupled services and RESTful APIs.
* Microservices architecture using **containerization** (Docker) and **orchestration** (Kubernetes or OpenShift) to enable scalability, modular deployment, and resilience.

**B. Layered Structure:**

* **Presentation Layer**:
  + Mobile App (built with **Flutter** for cross-platform compatibility or React Native).
  + Admin Web Portal (Angular/React with TailwindCSS or Material UI).
* **Application Logic Layer**:
  + Modular service units for:
    - Leave Management
    - Time & Attendance
    - Performance Appraisal
    - Payroll & Benefits
  + Built using Node.js (Express) or Python (Django/Flask).
* **Integration Layer**:
  + **API Gateway** (e.g., Kong or AWS API Gateway) for routing, throttling, and authentication.
  + Integration connectors to:
    - Existing **Legacy HRMIS** (if any)
    - **Biometric Devices** (ZKTeco, Suprema, etc.)
    - **ERP modules** (e.g., SAP, Oracle, or custom donor platforms)
* **Data Layer**:
  + **Primary DB**: PostgreSQL or MySQL with schema normalization, row-level locking, and full ACID compliance.
  + **Caching**: Redis/Memcached for user sessions and real-time status.
  + **Event Logging**: Elasticsearch for centralized log monitoring.

**C. Security & Redundancy:**

* API tokenization, request throttling, and audit trails.
* Asynchronous background workers (Celery, RabbitMQ) for syncing biometric and mobile offline data.
* **Disaster Recovery Plan** with automated backup to remote cloud storage.