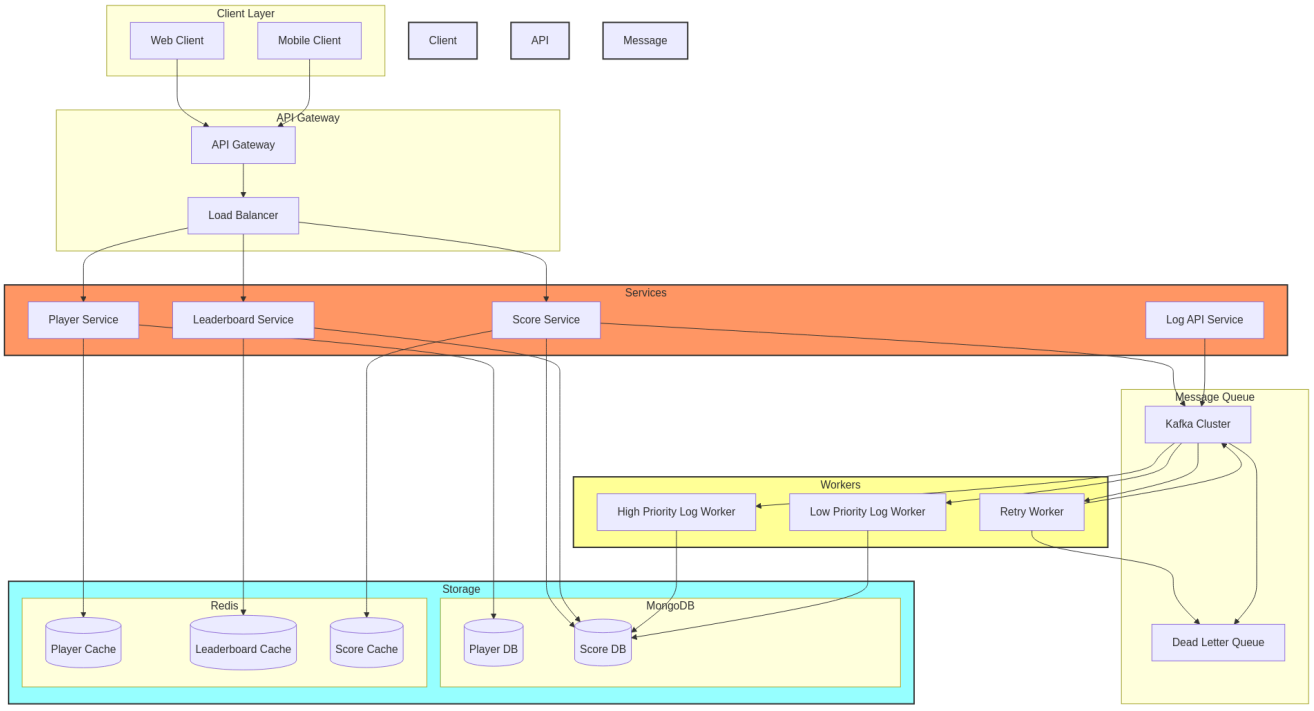
High Level - System Overview

System Overview:
This is a distributed gaming platform architecture designed for high scalability and reliability. The system handles player management, score tracking, and leaderboard functionality with real-time updates.



Why It's Stateless:

1. **Service Layer:**
   - All services (Player, Score, Leaderboard) are stateless
   - No session data stored in services
   - All state is managed by databases and caches
   - Services can be horizontally scaled without coordination
2. **Data Management:**
   - State is externalized to MongoDB and Redis
   - Services only contain business logic
   - No local state persistence
   - Consistent behavior across instances

Scalability Features:

1. **Horizontal Scaling:**
   - Services can be replicated
   - Load balancer distributes traffic

    - No shared state between instances
    - Independent scaling of components
2. **Data Layer Scaling:**
    - MongoDB sharding for data distribution
    - Redis cluster for caching
    - Kafka partitions for message processing
    - Separate read/write operations
3. **Performance Optimizations:**
    - Caching layer for frequent reads
    - Message queue for async processing
    - Worker separation for different priorities
    - Dead letter queue for error handling
4. **Resilience:**
    - Stateless services can fail without data loss
    - Message queue ensures data persistence
    - Retry mechanisms for failed operations
    - Cache fallback to database

This architecture allows for:

- Independent scaling of components
- High availability
- Fault tolerance
- Real-time performance
- Easy maintenance and updates