Notification System Architecture

Data Flow Legend

- ———— Synchronous API
- – – – Async (Queue)
- ········ WebSocket
- ———— External API

React Native Mobile App

React Web App

1. Create Notification

API Gateway + Load Balancer

8. Real-time Updates

2. Process

Notification API Service

WebSocket Server

9. Get User Tokens

Redis Cache (User Tokens)

4. Enqueue Delivery Jobs
Notification

Message Queue (RabbitMQ/SQS)

5. Dequeue

Push Worker

Email Worker

6. Send Push

FCM/APNS (Push)

7. Update Status

6. Send Email

SendGrid (Email)

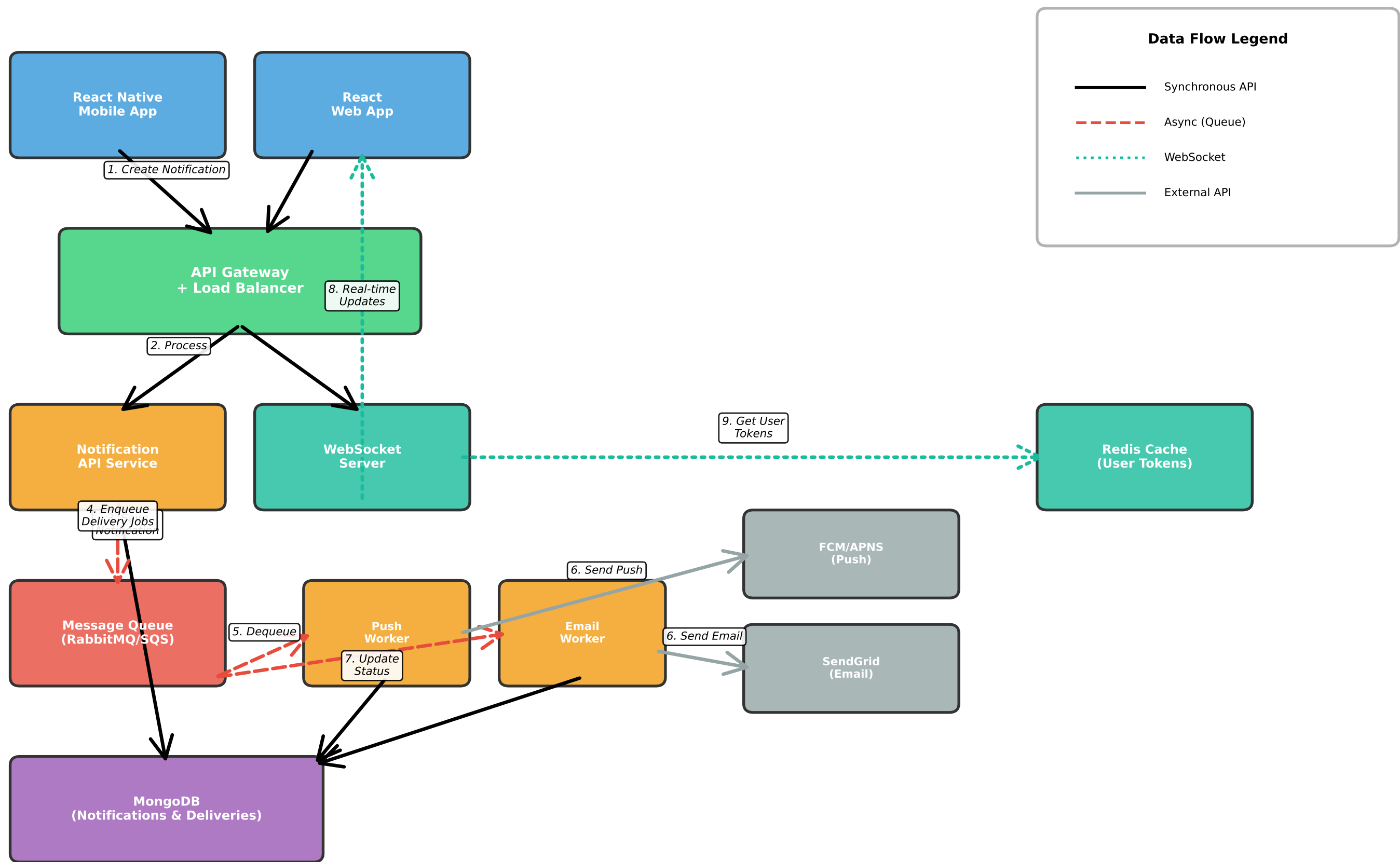**MongoDB (Notifications & Deliveries)**

Key Design Features:

- Async processing via Message Queue prevents API blocking
- WebSocket server enables real-time updates for web clients
- Workers handle retries independently for each channel
- Redis caches user push tokens for fast lookup
- MongoDB stores notification + per-channel delivery status
- Horizontal scaling: Add more workers for traffic spikes