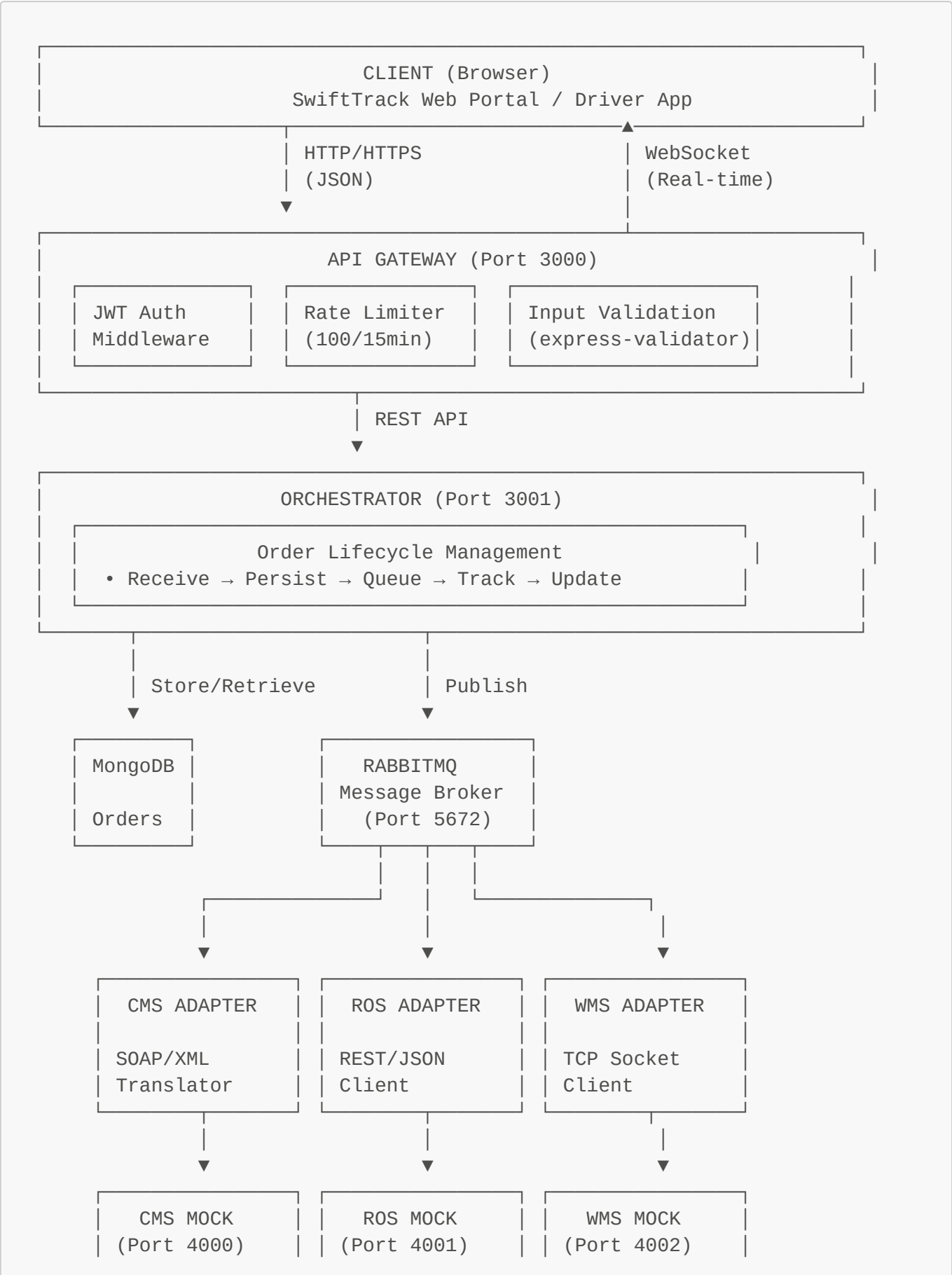
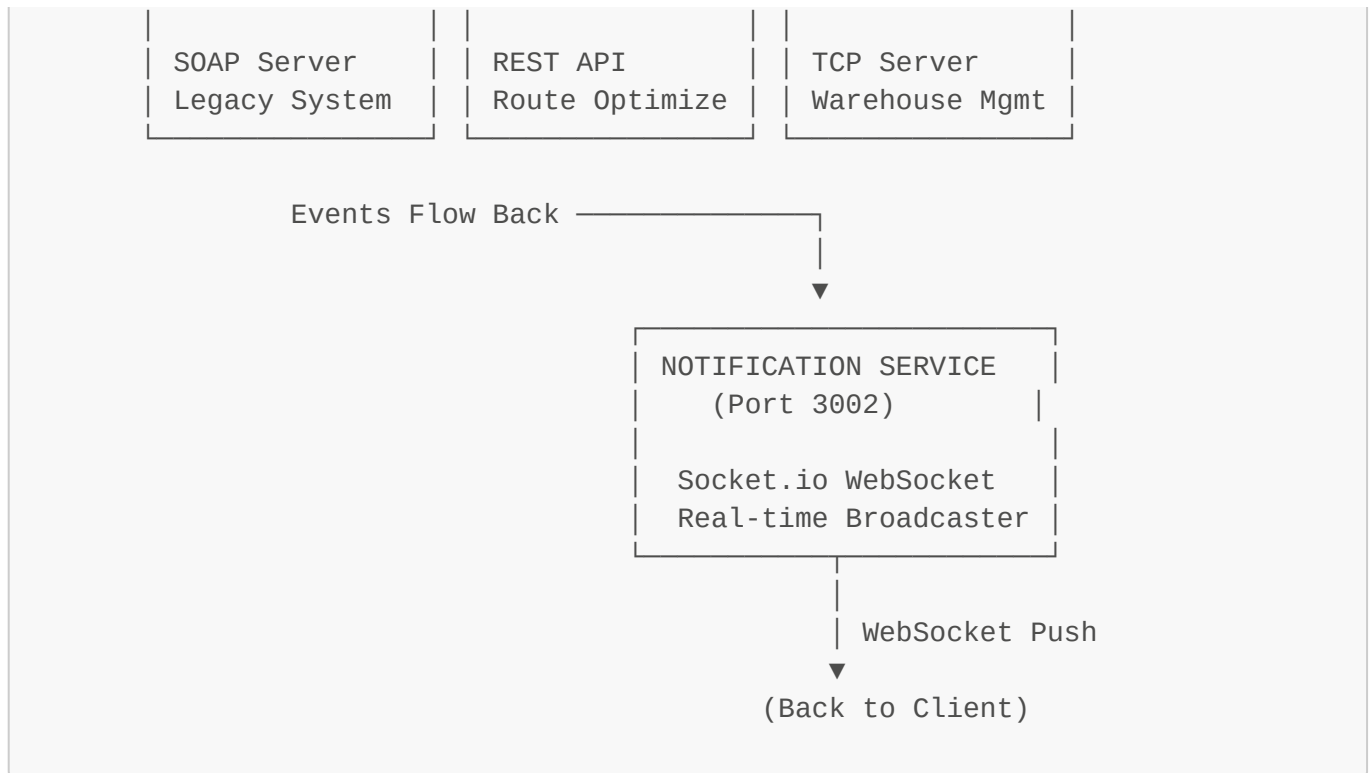


SwiftLogistics - System Architecture Diagrams

1. High-Level System Architecture





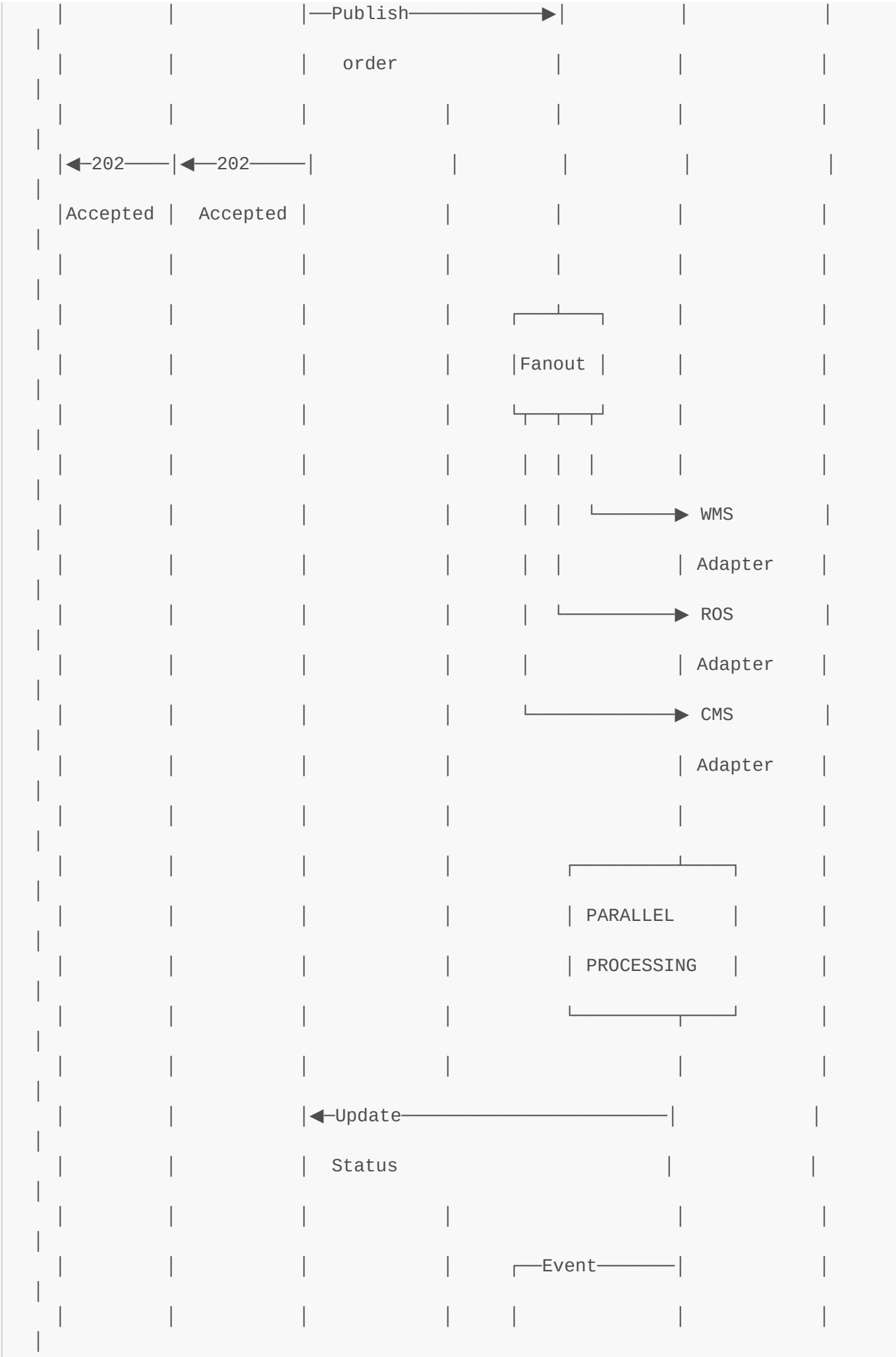
2. Order Submission Flow (Sequence)

```

graph LR
    subgraph Client
        direction TB
        C1[Client Notification]
        C2[Client]
    end
    subgraph Gateway
        direction TB
        G1[Gateway]
        G2[Client]
    end
    subgraph Orchestrator
        direction TB
        O1[Orchestrator]
        O2[JWT]
    end
    subgraph MongoDB
        direction TB
        M1[MongoDB]
        M2["(RECEIVED)"]
    end
    subgraph RabbitMQ
        direction TB
        R1[RabbitMQ]
    end
    subgraph Adapters
        direction TB
        A1[Adapters]
    end

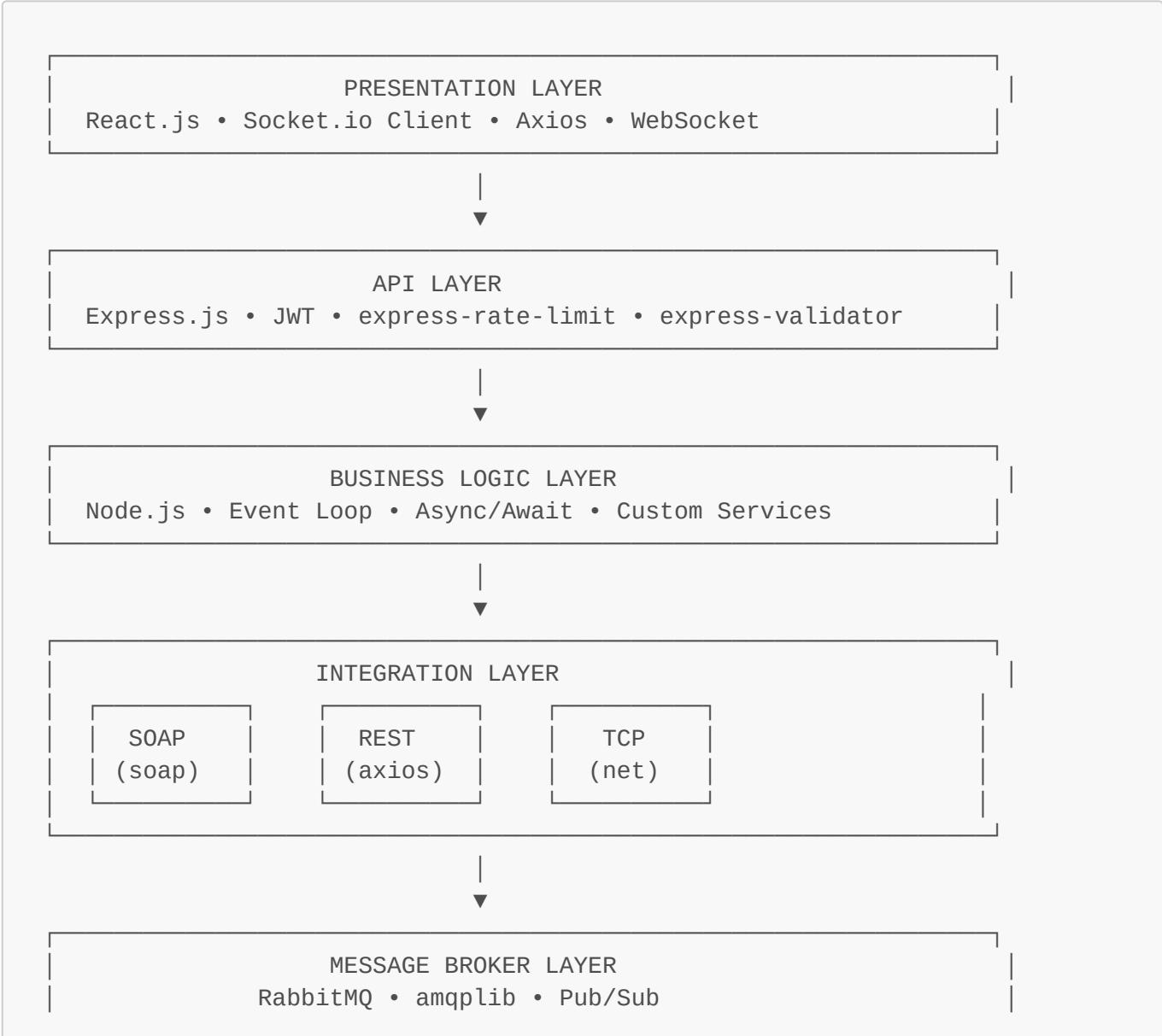
    C1 -- "POST" --> G1
    G1 -- "Validate" --> O1
    O1 -- "JWT" --> G2
    G2 -- "Forward" --> O2
    O2 -- "Save" --> M1
    M1 -- "(RECEIVED)" --> R1
    R1 -- "OK" --> M2
    M2 -- "OK" --> O2
    O2 -- "OK" --> G2
    G2 -- "OK" --> C2

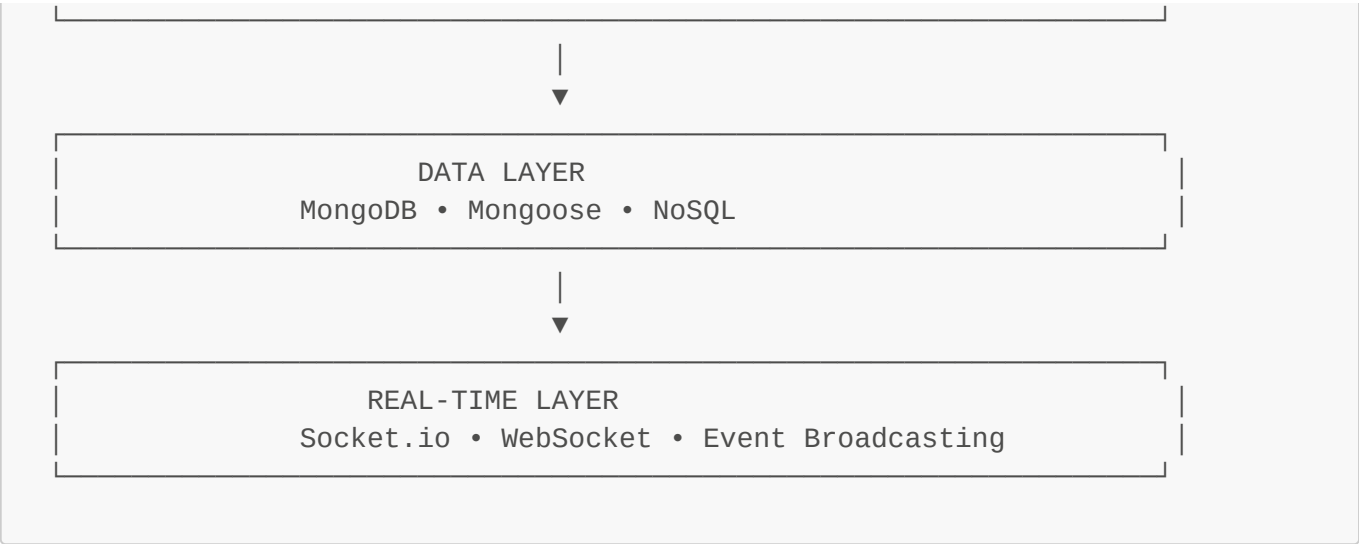
```



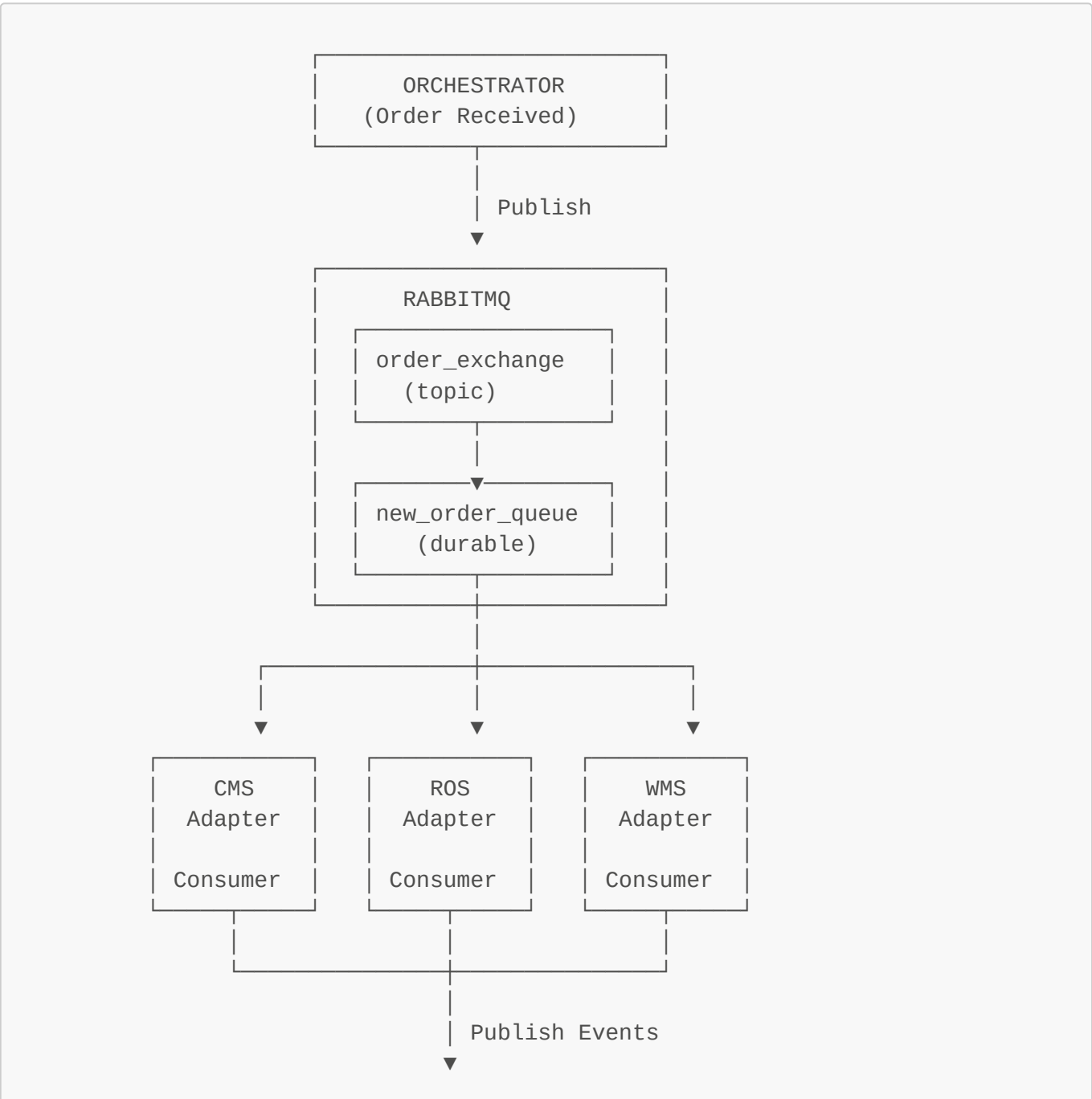


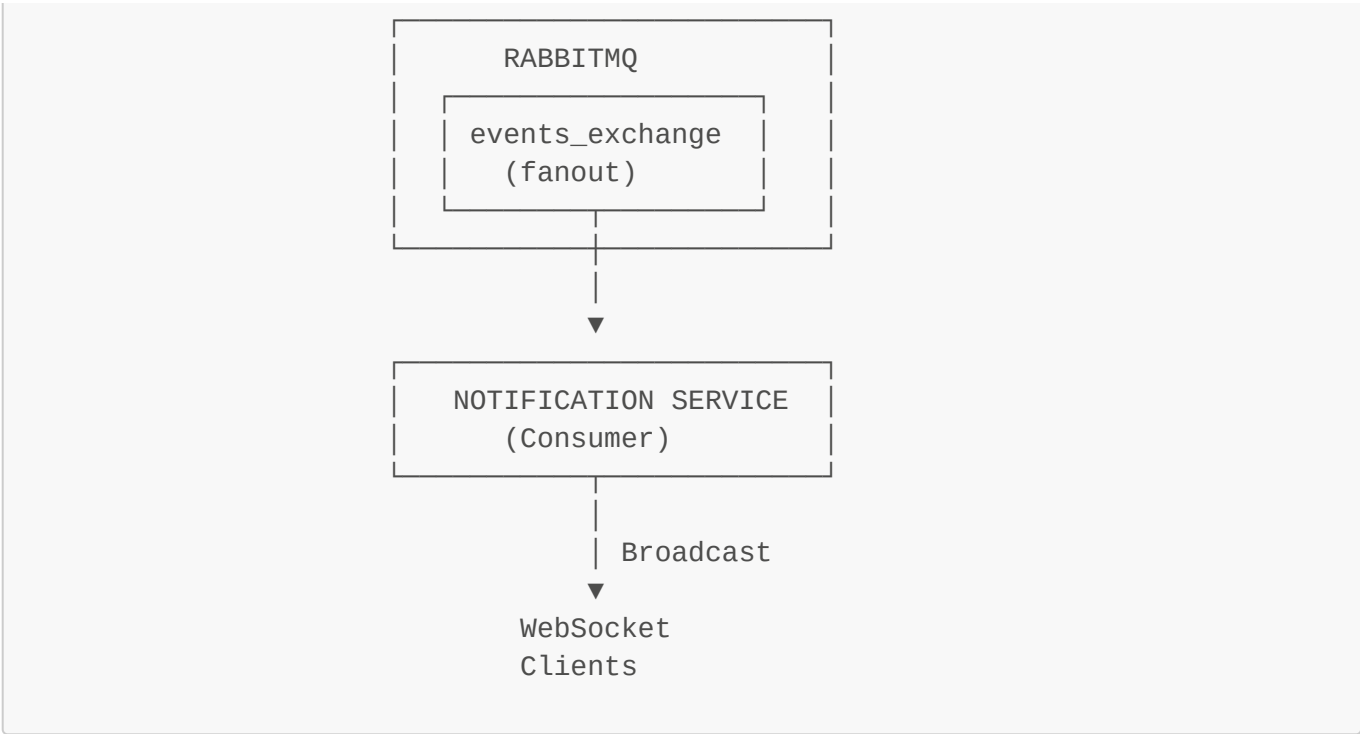
3. Technology Stack Mapping



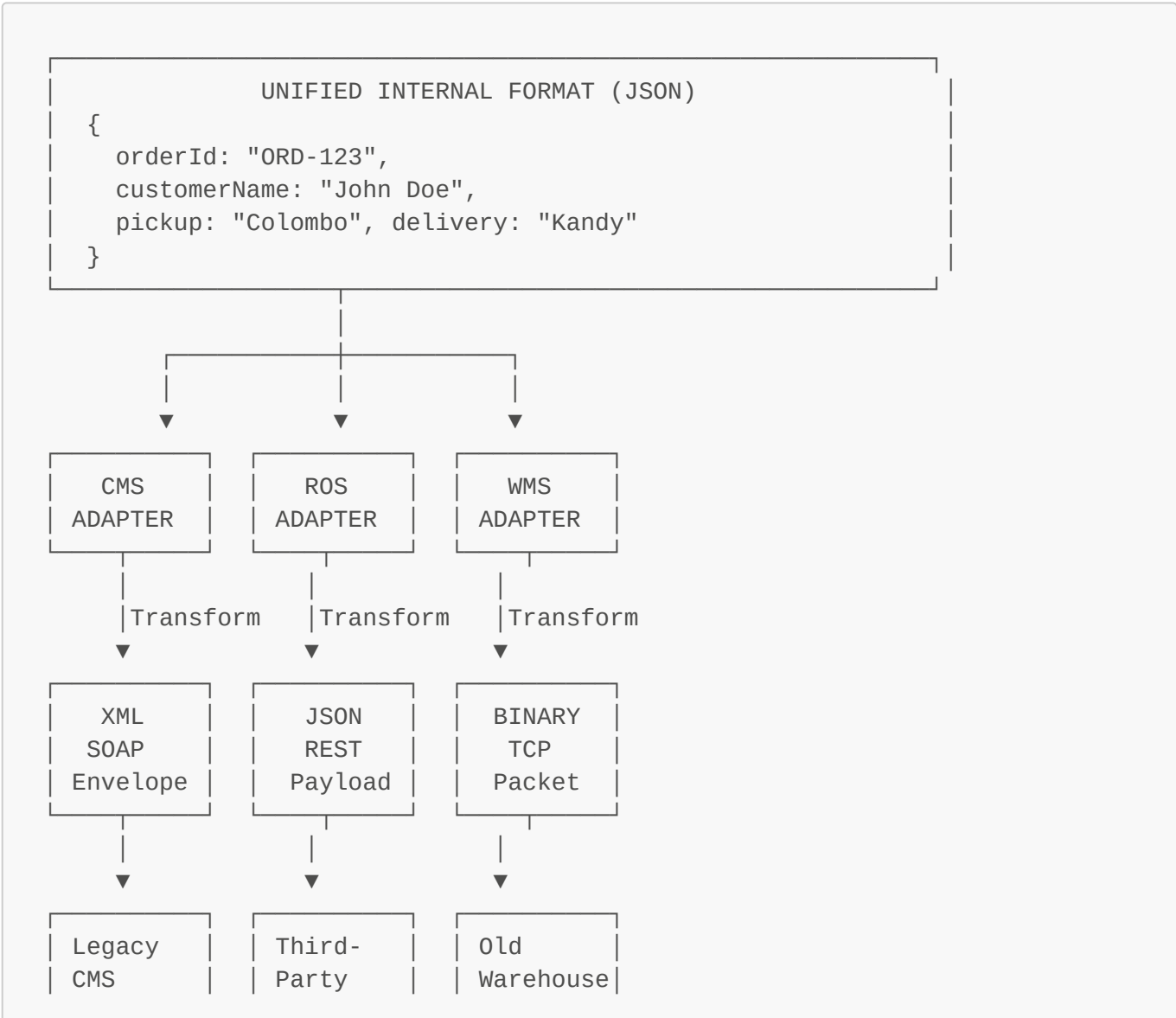


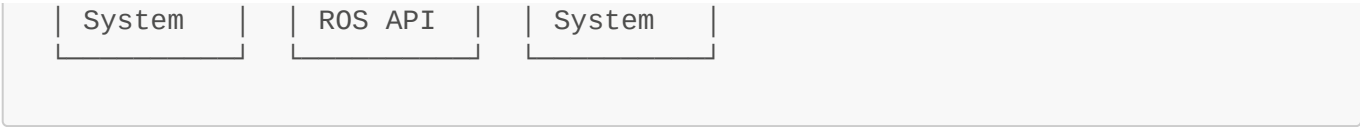
4. Message Queue Pattern





5. Adapter Pattern Implementation





Legend

→

Synchronous HTTP Request (blocking)

- ->

Asynchronous Message Queue (non-blocking)

~~~>

WebSocket Real-time Push

==>

Database Operation

Port Summary

| Service      | Port  | Protocol  | Purpose            |
|--------------|-------|-----------|--------------------|
| API Gateway  | 3000  | HTTP/REST | Client entry point |
| Orchestrator | 3001  | HTTP/REST | Transaction mgmt   |
| Notification | 3002  | WebSocket | Real-time updates  |
| CMS Mock     | 4000  | SOAP/XML  | Legacy system sim  |
| ROS Mock     | 4001  | REST/JSON | Route service sim  |
| WMS Mock     | 4002  | TCP       | Warehouse sim      |
| MongoDB      | 27017 | MongoDB   | Database           |
| RabbitMQ     | 5672  | AMQP      | Message broker     |
| RabbitMQ UI  | 15672 | HTTP      | Admin interface    |