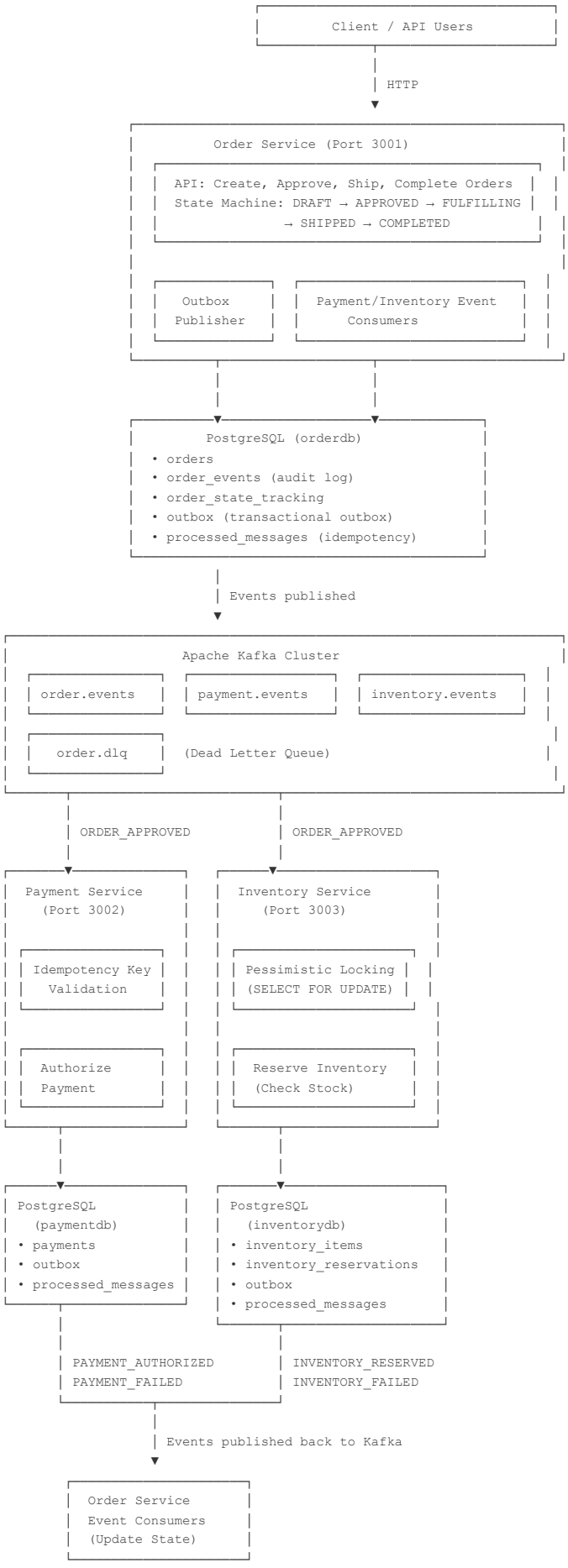
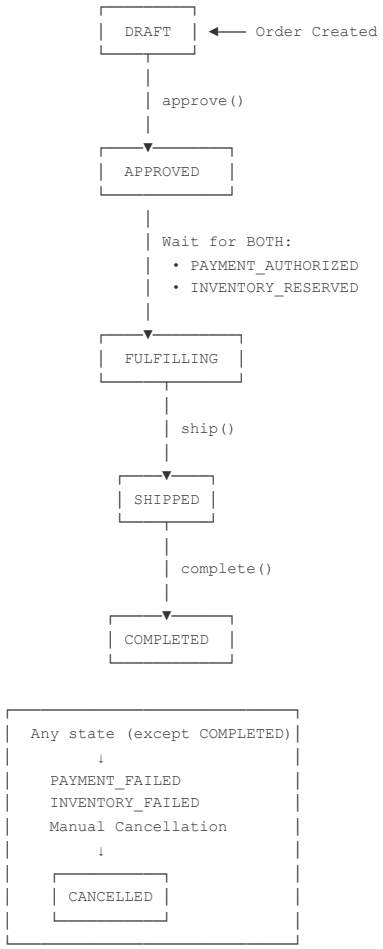
# OrderFlow Architecture Diagrams

## Table of Contents

## System Architecture



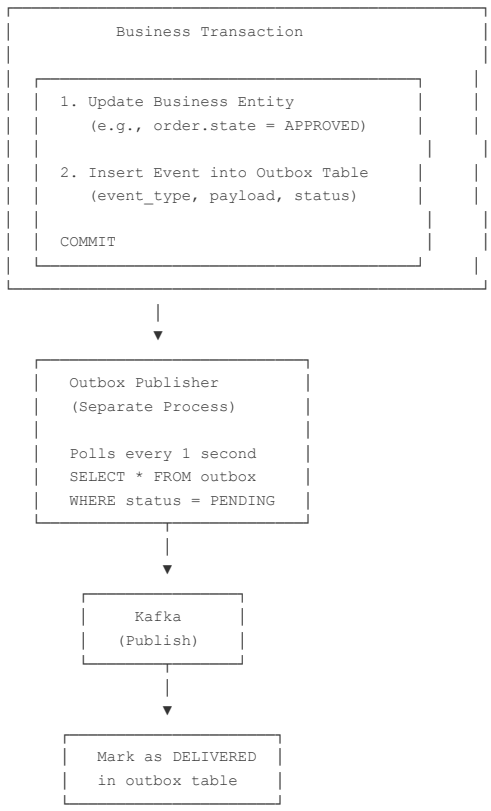
## Order State Machine



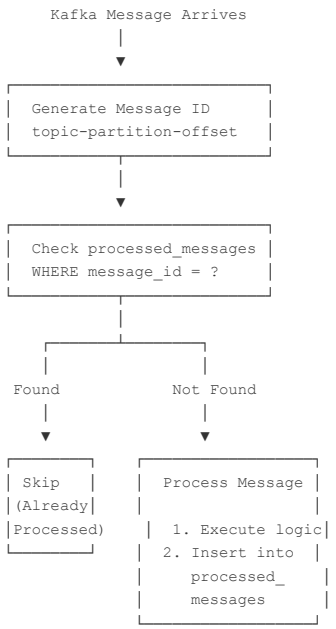
## Event Flow - Happy Path

## Production Patterns

### 1. Transactional Outbox Pattern




### 2. Idempotent Consumer




### 3. Out-of-Order Event Handling

> **Order State Tracking Table:**

```
┌──────────┬────────────────────┬────────────────────┐
│ order_id │payment_authorized│inventory_reserved│
├──────────┼────────────────────┼────────────────────┤
│ 123-abc  │       false        │       false        │
└──────────┴────────────────────┴────────────────────┘
```

```
Scenario A: Payment arrives first
│
├─► PAYMENT_AUTHORIZED arrives
│   └─► Update: payment_authorized = true
│   └─► Check: inventory_reserved? NO → Wait
│
└─► INVENTORY_RESERVED arrives
    └─► Update: inventory_reserved = true
    └─► Check: BOTH true? YES → Transition to FULFILLING

Scenario B: Inventory arrives first
│
├─► INVENTORY_RESERVED arrives
│   └─► Update: inventory_reserved = true
│   └─► Check: payment_authorized? NO → Wait
│
└─► PAYMENT_AUTHORIZED arrives
    └─► Update: payment_authorized = true
    └─► Check: BOTH true? YES → Transition to FULFILLING

Result: Order progresses correctly regardless of event order!
```

## Key Metrics & Monitoring

| Category | Metrics |
| --- | --- |
| **Service Metrics** | • Request Rate (orders/sec)<br>• Request Latency (p50, p95, p99)<br>• Error Rate (% failed requests)<br>• Order State Distribution |
| **Kafka Metrics** | • Consumer Lag (messages behind)<br>• Message Throughput (msgs/sec)<br>• Failed Messages → DLQ count |
| **Database Metrics** | • Outbox Queue Depth (pending events)<br>• Query Latency<br>• Connection Pool Usage<br>• Lock Wait Time (inventory) |

## Failure Scenarios

| Failure | Impact | Recovery |
| --- | --- | --- |
| Kafka Down | Events buffered in outbox | Outbox holds events, auto publishes when Kafka back |
| Payment Service Down | Orders stuck in APPROVED state | Consumer catches up from offset |
| Inventory Out of Stock | Order cancelled | INVENTORY_FAILED event triggers cancellation |
| Payment Gateway Fails | Order cancelled | PAYMENT_FAILED event triggers cancellation |
| Database Connection Lost | Service unavailable | Restart service, reconnect |

## Technology Stack

| Layer | Technologies |
| --- | --- |
| **Application Layer** | • TypeScript + Node.js 20+<br>• Fastify (HTTP framework)<br>• Zod (Schema validation) |
| **Messaging Layer** | • Apache Kafka (Event streaming)<br>• kafkajs (Client library)<br>• Idempotent producers |
| **Data Layer** | • PostgreSQL 15 (ACID transactions)<br>• pg (Client library)<br>• Per-service databases |
| **Observability Layer** | • Pino (Structured logging)<br>• prom-client (Metrics)<br>• Correlation IDs |
| **Infrastructure** | • Docker + Docker Compose<br>• GitHub Actions (CI/CD)<br>• Jest (Testing) |

---