

# Backend Engineering Bootcamp

## Microservices Study Cases

### Project Guidelines

- Teams of 5 students
- Mandatory use of Go for all services
- Infrastructure as Code (Docker required)

## 1. Traveloka Clone (Booking Platform)

### Functional Requirements

- **User Management:**
  - Registration/login with JWT
  - Profile editing
- **Booking Flow:**
  - Search flights/hotels with filters
  - Reserve with temporary holds
  - Payment processing (mock)
- **Admin Features:**
  - Inventory management
  - Booking analytics

### Non-Functional Requirements

Category	Requirement
Performance	Search results returned in <500ms
Availability	99% uptime for booking service
Observability	All APIs instrumented with metrics
Security	Password hashing with bcrypt
Data Consistency	Booking conflicts prevented

## 2. Instagram Clone (Social Network)

### Functional Requirements

- **Content:**
  - Image upload with thumbnails
  - Post captioning/tagging
- **Social Graph:**
  - Follow/unfollow users
  - Timeline generation
- **Interactions:**
  - Like/comment functionality
  - Real-time notifications

### Non-Functional Requirements

Category	Requirement
Scalability	Handle 1000 concurrent timeline requests
Data Freshness	Feed updates within 30 seconds
Durability	Zero image data loss
Cost	Optimize cloud storage usage
Extensibility	API versioning support

## 3. WhatsApp Clone (Messaging App)

### Functional Requirements

- **Messaging:**
  - 1:1 and group chats
  - Message status tracking
- **Presence:**
  - Online status indicators
  - Last seen timestamps
- **Media:**
  - Image/document sharing
  - End-to-end encryption (mock)

## Non-Functional Requirements

Category	Requirement
Latency	Messages delivered in <300ms
Reliability	99.9% message delivery rate
Ordering	Guaranteed FIFO in conversations
Retention	30-day message history
Compliance	GDPR-ready deletion workflows

## Technical Constraints

- All services must expose REST/WebSocket APIs
- Minimum 3 microservices per project
- Required components:
  - caching layer
  - monitoring
  - logging