Dhirubhai Ambani University

Technology

Formerly DA-IICT

IT457 Cloud Computing

Assignment 1

Architecting cloud requirements based on applications

Neel Patel (Group 5)

202201494

Airbnb Mobile Backend - N-Tier Architecture

Airbnb is a global platform for booking stays, experiences, and managing host–guest interactions. It requires a scalable, secure, and highly available cloud backend to handle millions of users, real-time bookings, payments, and recommendations.

Step 1 : Define the Business or Organization Problem

Challenge: Automate the online property rental marketplace connecting hosts with guests globally, enabling seamless booking, payment processing, and property management.

Main Function/Purpose: Enable property owners to list their spaces and travelers to discover, book, and pay for accommodations worldwide through mobile and web platforms.

Step 2 : Define Necessary Business Processes

Sub Functions

SF1 : Guest Interaction & Booking

- User Interface & Presentation
- Search and browse properties
- View property details, photos, reviews
- Make reservations and payments
- Manage bookings and communicate with hosts

SF2: Host Property Management

- Application Logic
- List properties with details and pricing
- Manage availability calendars
- Process booking requests
- Handle guest communications
- Manage pricing and promotions

SF3: Business Operations & Data Management

- Data Storage & Manipulation
- User profiles (guests and hosts)
- Property information and media
- Booking and payment records
- Reviews and ratings system
- Geographic and mapping data

Step 3 : Software Requirements

Use Case : Guest booking a property on Airbnb mobile app

Key Functions & Features :

- Type in search criteria (location, dates, guests, filters)
- Present property listings with photos, pricing, ratings
- Display property details including amenities, location, host info
- Provide booking interface with date selection and pricing calculation
- Process secure payments through integrated payment gateway
- Enable messaging system between guests and hosts
- Send booking confirmations and notifications
- Manage user profiles and preferences
- Handle reviews and ratings post-stay

Step 4 : Software Architecture

Layer 1 : Thin Client (Mobile/Web Interface)

Purpose : Enable GUI for clients on mobile devices and web browsers

- iOS/Android mobile applications
- Responsive web interface
- Real-time messaging interface
- Map integration and location services

Layer 2 : Presentation Layer

Purpose: Present necessary information and create GUI for clients Functions:

- Property catalog display with filtering
- Booking interface and calendar integration
- User profile management interface

- Search results visualization
- Payment processing interface
- Messaging and notification system

Layer 3 : Application Logic Layer

Purpose: Perform core business logic operations Components:

Guest Logic

- Search and recommendation algorithms
- Booking validation and processing
- Payment processing coordination
- Review and rating management

Host Logic

- Property listing management
- Availability calendar synchronization
- Pricing optimization algorithms
- Revenue tracking and analytics

Platform Logic

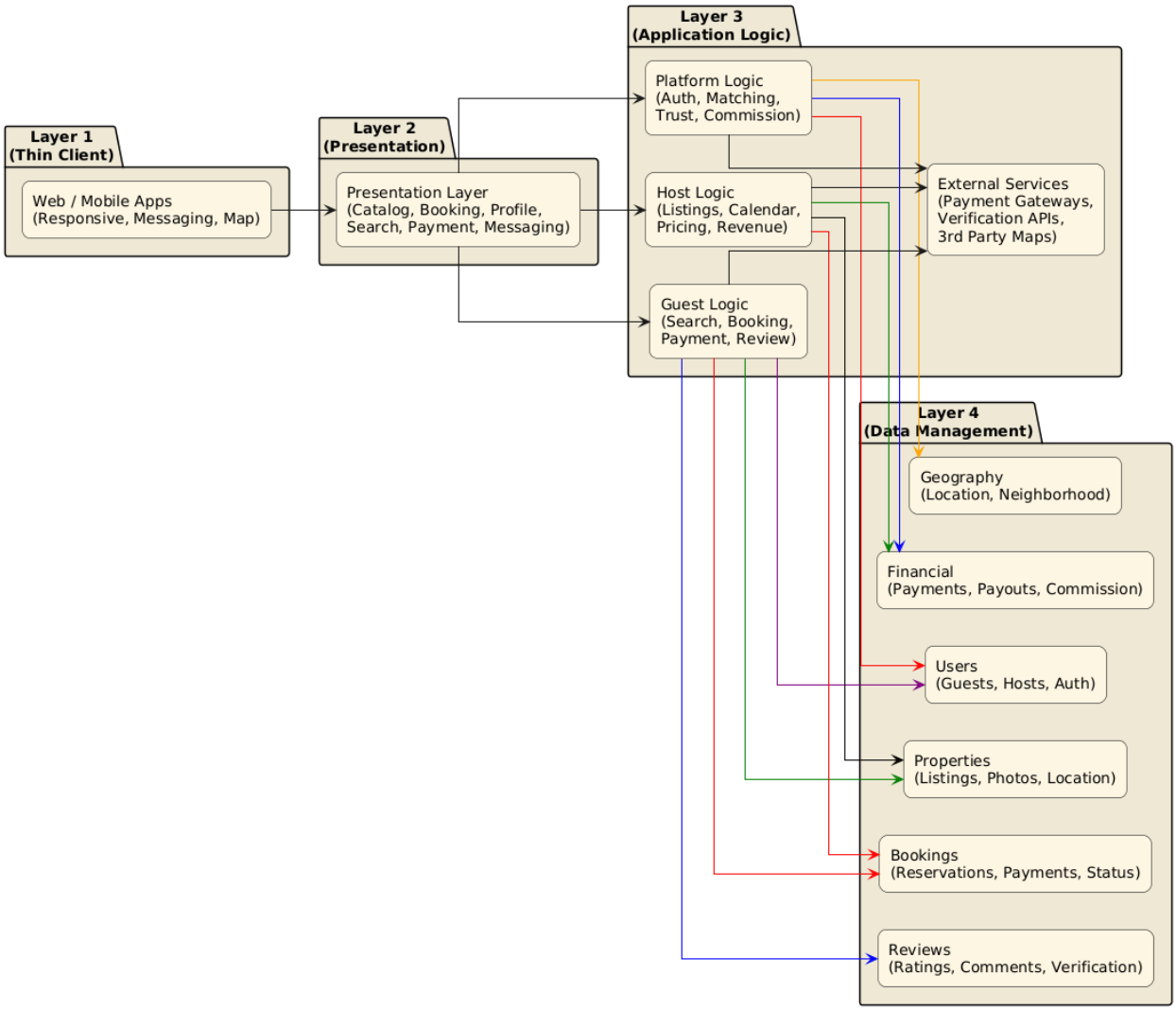
- User authentication and authorization
- Matching algorithms (guests to properties)
- Trust and safety verification
- Commission calculation and processing

Layer 4 : Data Management Layer

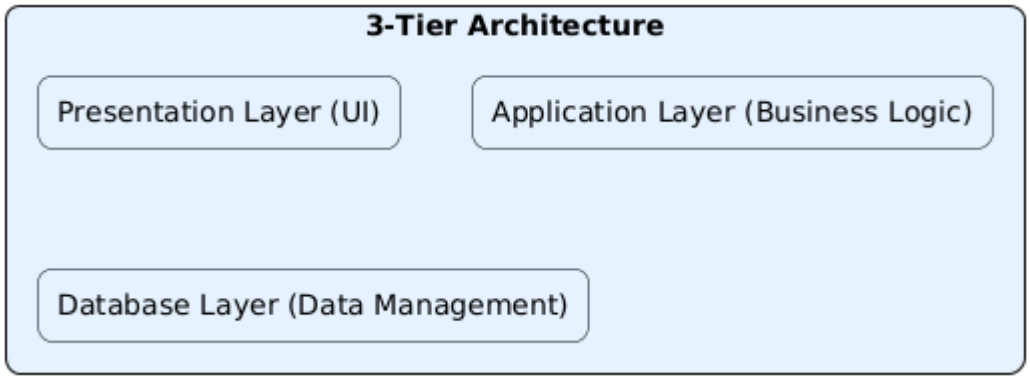
Purpose: Manipulate and share data across the platform Data Entities:

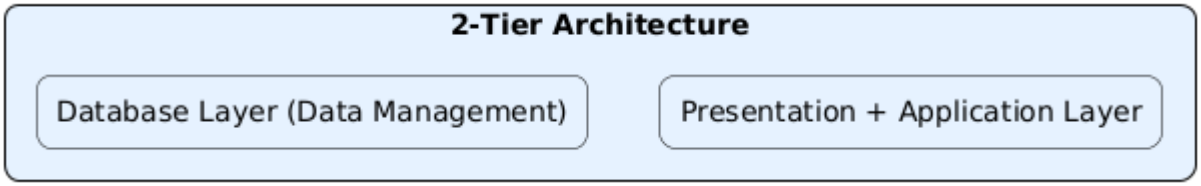
- **Users:** Guest profiles, host profiles, authentication data
- **Properties:** Listings, amenities, photos, location data
- **Bookings:** Reservations, payments, dates, status
- **Reviews:** Ratings, comments, verification status
- **Geography:** Location data, neighborhood information
- **Financial:** Payment records, payouts, commissions

Software Architecture Diagram - Airbnb

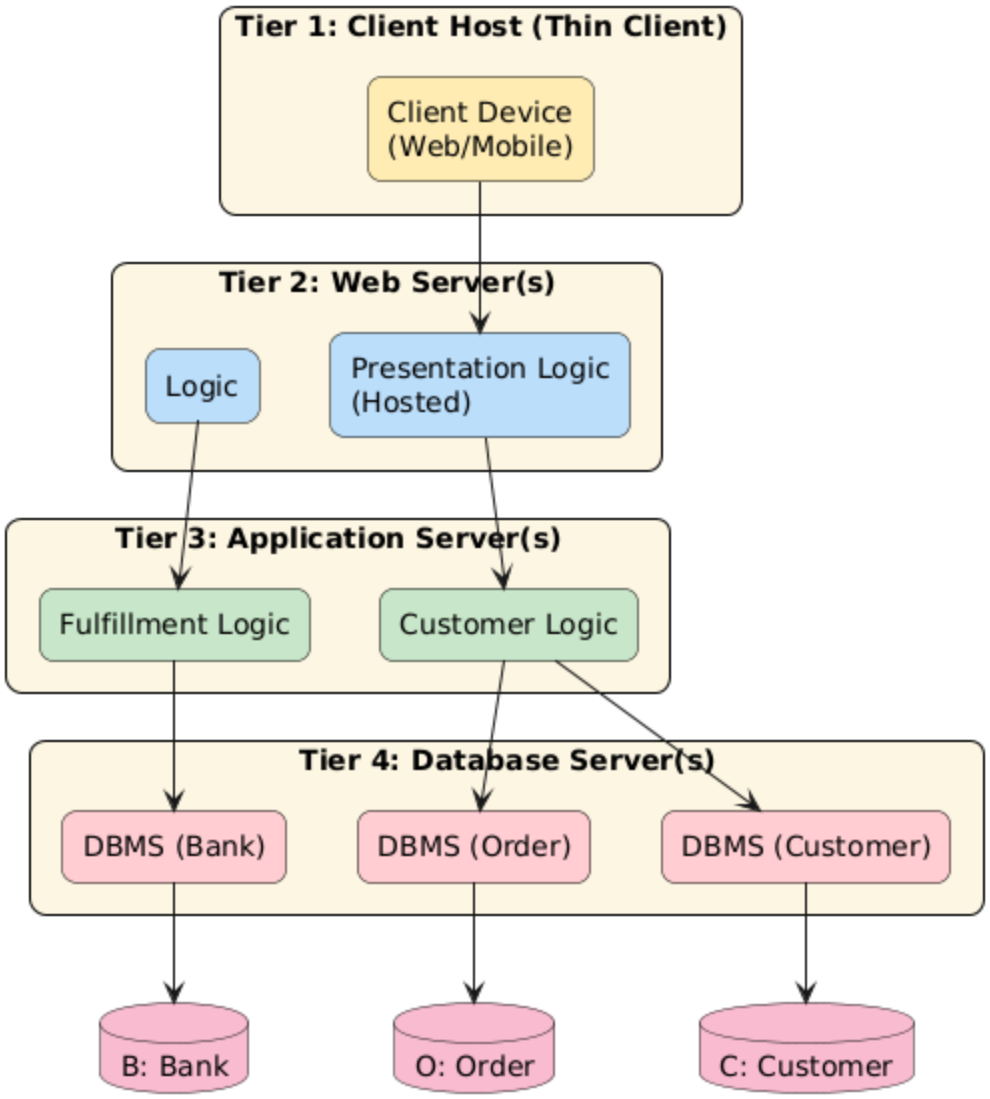


Step 5


3-Tier Architecture
Presentation Layer (UI)
Application Layer (Business Logic)
Database Layer (Data Management)


2-Tier Architecture
Database Layer (Data Management)
Presentation + Application Layer

Hardware Architecture - 4 Tier Client-Server (Client + 3 Server Tiers)



Technical Implementation Notes

Inter-layer Communication Protocols:

- **Client ↔ Web Servers:** HTTPS, WebSocket (for real-time messaging)
- **Web ↔ Application:** REST APIs, GraphQL
- **Application ↔ Database:** SQL, NoSQL query protocols
- **Common Gateway Interface:** RESTful APIs with JSON

Cloud Infrastructure Requirements:

1. ***Database Infrastructure:***
    - Distributed database systems for global scalability
    - Data replication across multiple regions
    - Caching layers (Redis/Memcached) for performance

2. ***Network Infrastructure:***
    - Global Content Delivery Network (CDN)
    - Auto-scaling load balancers
    - High-availability network architecture
    - API rate limiting and security