



# Multi-Tenant Rickshaw Booking System: An OS Perspective

This presentation explains core OS concepts using a rickshaw booking platform. We explore scheduling, concurrency, and resource management in a real-world context. Understand practical OS applications for multi-tenant environments and concurrent systems.

# System Architecture: Multi-Tenancy Explained

## Multiple Tenants

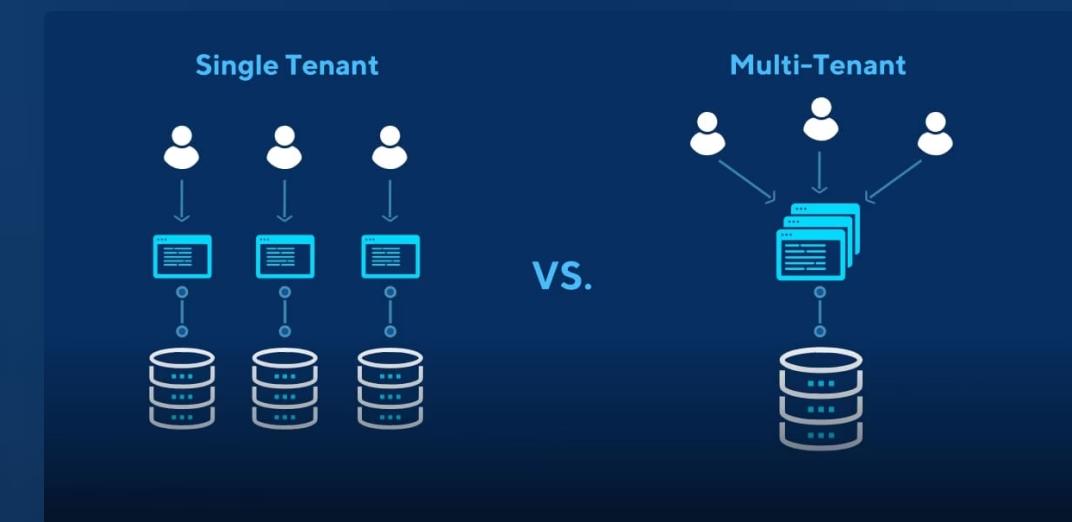
Separate rickshaws share one booking system platform.

## Resource Isolation

Each tenant manages its own rickshaws, drivers, and bookings securely.

## Shared Infrastructure

Common servers, databases, and APIs optimize resource usage and cost.



# Process Scheduling: Balancing Tenant Demands

## FCFS Scheduling

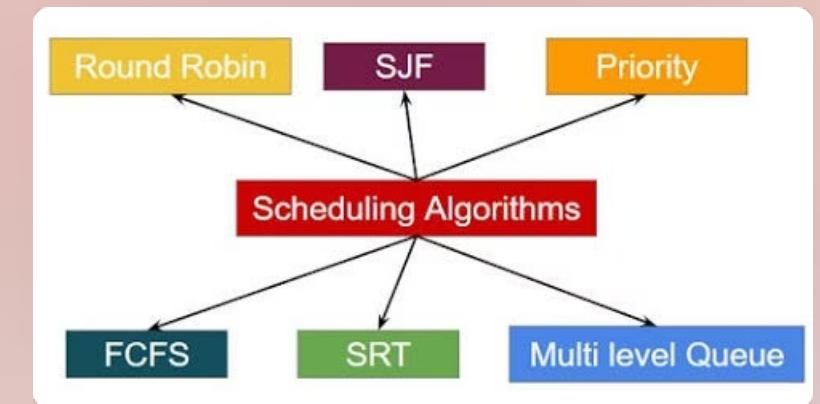
First-come booking requests handled in order received.

## Priority Scheduling

Premium tenants or urgent requests get faster processing.

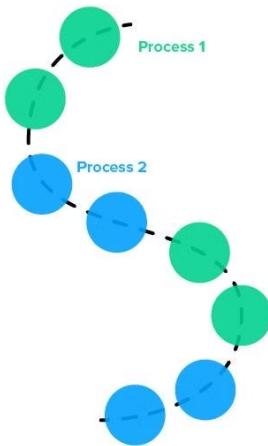
## Round Robin

Fair time slices ensure equal opportunity for all tenants.



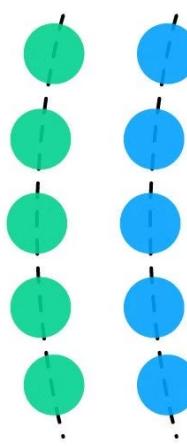
# Concurrency and Mutual Exclusion: Booking Race Conditions

Concurrency



vs

Parallelism



## Race Conditions

Conflicts arise when users book the same rickshaw simultaneously.

## Mutual Exclusion

Semaphores and mutexes protect data consistency during booking.

## Atomic Operations

Booking confirmation is completed as a single indivisible step.

# Deadlock Prevention: Avoiding Rickshaw Gridlock

## Deadlock Conditions

Circular wait and resource holding cause booking standstills.

## Resource Ordering

Assigning resource request sequences prevents circular waits.

## Timeout Mechanisms

Bookings expire after waiting too long, freeing rickshaw resources.



# Memory Management: Isolating Tenant Data

## Virtual Memory

Each tenant accesses isolated memory addresses for security.

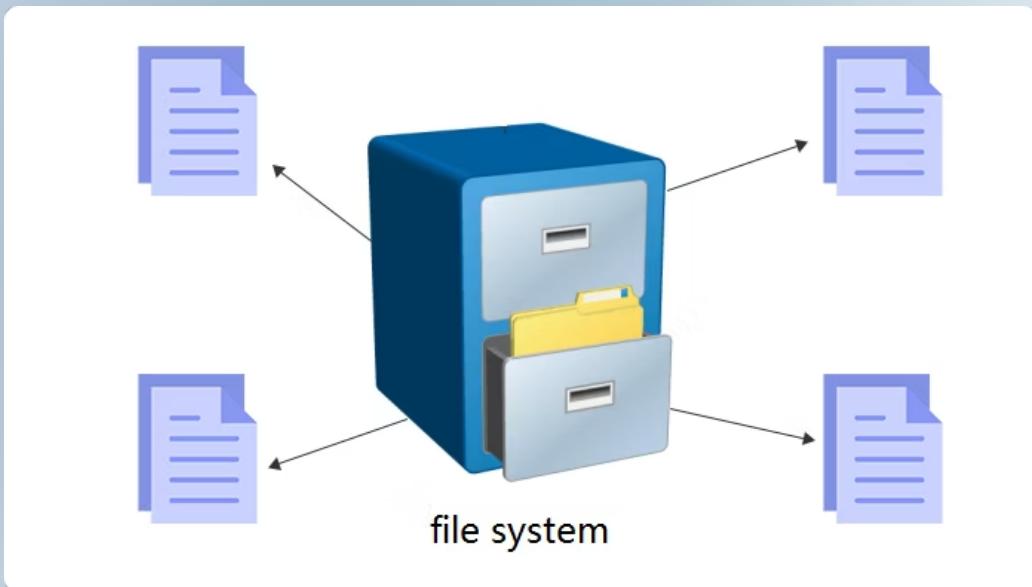
## Dynamic Allocation

Memory is assigned on demand for booking data and operations.

## Paging and Segmentation

Efficient memory use with page swapping and segmented data handling.

# File System Management: Booking Logs and Audit Trails



**Tenant Directories**  
Separate folders isolate tenant booking and operational files.



## Access Control Lists

Permissions protect data from unauthorized users and processes.

## Journaling File System

Ensures data integrity even in case of system crashes.

# Security: Protecting Against Unauthorized Access

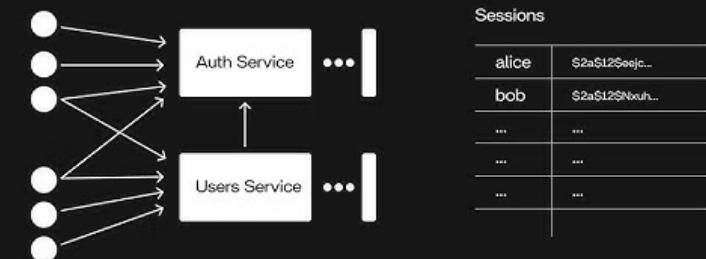
**User Authentication**  
Verifies identities of passengers and rickshaw drivers securely.



**Authorization**  
Controls which resources user roles can access or modify.

**Data Encryption**  
Secures booking info during transmission over networks.

## An Authentication System



# OS-Level Resource Monitoring and Management

## CPU Utilization

Track server load to optimize job scheduling and responsiveness.

## Memory Usage

Monitor tenant memory to prevent resource starvation and crashes.

## I/O Operations

Enhance disk access speed for quick booking data retrieval.

# Thank You.

**Prateek Deshmukh (RA2311047010111)**

**Dibyaday Betal (RA2311047010114)**

**Sivasai Bimavaram (RA2311047010064)**

**Akshita Singh Tyagi (RA2311047010075)**