Indian Institute of Technology, Roorkee

CSC 303: Computer Networks

Requirements and Specification Document

BlockRide – Decentralized Ride Sharing

Collaborators (Team 10)

Aryan Choudhary	23115024
Parth Baranwal	23114076
Rohan Gupta	23114088
Shiv Shakti Kumar	23114091
Siddharth Gupta	23114093
Vishesh Gupta	23114107

8th October 2025

BlockRide

Requirements and Specification Document

Date: 08-10-2025
Version: 0.1

1 Project Abstract

BlockRide will be a decentralized ride sharing platform that enables riders and drivers to discover each other, exchange intent, and confirm rides using a blockchain anchored registry and decentralized storage. Users will authenticate using a Web3 wallet, store minimal user and ride intent data in content addressed storage, and communicate via real-time messaging to confirm rides. The system will prioritize privacy, transparency, and user control while maintaining a familiar web experience.

2 Project Rationale and Applications

2.1 Motivation

Urban mobility frequently suffers from opaque pricing, high intermediation fees and limited user control over data. Decentralized primitives namely, content addressed storage and smart contracts allow verifiable data integrity, transparent governance, and lower platform overhead. BlockRide will leverage these to create a privacy respecting, open ride sharing marketplace where participants control their data and can interact peer-to-peer with light coordination.

2.2 Application and Prospective Customer

- **General audience**: Commuters seeking cost-effective, privacy respecting ride sharing; drivers wanting flexible matching without high intermediary commissions; privacy conscious users who prefer data minimization.
- **Prospective customer**: “MetroMobility Cooperative” (dummy customer) piloting BlockRide for office commuters within a 15 km radius.

3 Competitive Landscape

The ride-sharing market exhibits significant variation across countries and regulatory environments. Our primary competitors fall into three categories: **centralized ride hailing platforms** (Uber, Ola, Lyft), **carpool focused services** (BlaBlaCar), and emerging **local ride sharing networks**. These incumbents have established substantial market presence through years of operational refinement and capital investment.

3.1 Competitive Analysis

Strengths of Existing Solutions: The dominant players in this space have built formidable competitive forts. Their most significant advantage is **dense marketplace liquidity** powered by strong network effects i.e. more riders attract more drivers and vice versa, creating a self-reinforcing cycle. From a technical standpoint, these platforms offer **mature mobile experiences** with seamless in-app payments, comprehensive customer support infrastructure, and increasingly sophisticated safety features including real-time tracking, emergency assistance, and driver verification systems. Additionally, their **dynamic pricing algorithms** and intelligent matching systems have been refined over millions of rides, optimizing for efficiency and user satisfaction.

Weaknesses and Opportunities for Differentiation: Despite their market dominance, centralized platforms face growing criticism on several fronts. They typically charge **commission rates of 20–30%**, significantly eroding driver earnings while maintaining opaque fee structures that frustrate both sides of the marketplace. The **centralized control of user data** and unilateral policy changes create trust issues and **platform lock-in effects**. Furthermore, **reputation and rating portability is nonexistent** i.e. a driver with thousands of five-star rides on one platform starts from zero on another, creating artificial switching costs and reducing competition.

3.2 BlockRide's Differentiation Strategy

BlockRide addresses these weaknesses through fundamental architectural choices. Our platform utilizes an **on chain registry of ride intent commitments** (stored as Merkle roots), providing cryptographic auditability without sacrificing user privacy. By implementing **content-addressed storage** for user profiles and ride data, we ensure data integrity and enable potential cross-platform interoperability that centralized competitors cannot offer.

The technical architecture supports **open, modular interfaces** that allow community operated matching algorithms and dispute resolution mechanisms to compete, rather than imposing a single algorithmic approach. This openness, combined with reduced infrastructure overhead from distributed systems, enables us to target **platform fees in the 5–10% range** a significant reduction that directly benefits both drivers and riders while still maintaining economic sustainability.

3.3 Barriers and Challenges

We acknowledge several significant competitive barriers. The **two-sided marketplace liquidity problem** is perhaps our most critical challenge i.e. riders won't use a platform without drivers, and drivers won't join without riders. Our go-to-market strategy addresses this through targeted geographic launches and driver incentive programs.

Regulatory compliance and safety infrastructure cannot be compromised, even in a decentralized system. We are designing safety features (identity verification, ride tracking, emergency protocols) that meet or exceed centralized platform standards, though implementation in a privacy preserving, decentralized context requires novel technical approaches.

Finally, **cryptocurrency UX friction** remains a real barrier to mainstream adoption. Wallet onboarding, gas fee management, and key security are unfamiliar concepts to most users. Our strategy includes progressive decentralization, starting with familiar

Web2 UX while gradually introducing blockchain benefits and exploring account abstraction solutions to minimize crypto-native complexity for end users.

3.4 Intellectual Property Considerations

The current patent landscape in ride sharing focuses primarily on matching algorithms, dynamic pricing mechanisms, and safety features, these are areas where patents are already held by major incumbents. However, the intersection of blockchain technology with ride sharing presents opportunities for novel patent positions, particularly around decentralized reputation systems, on-chain ride verification protocols, and privacy-preserving matching mechanisms. We are conducting ongoing patent prior art searches in these areas to identify potential filing opportunities that could create defensive IP positions.

4 System Requirements

4.1 System Context and Architecture

- Clients: Web browser (mobile) with Web3 wallet
- Smart Contract Layer: minimal contract to record registry CIDs (Content Identifiers)
- Decentralized Storage: IPFS (JSON documents)
- Real-Time Signaling: WebSocket backend
- Mapping & Geospatial: Web map SDK

4.2 Functional Requirements

FR-1 Wallet Authentication

FR-2 User Registration

FR-3 View Profile

FR-4 Create Driver Offer

FR-5 Create Rider Request

FR-6 Driver Suggestions for Rider

FR-7 Real-Time Confirmation

FR-8 Logout

FR-9 Error Handling

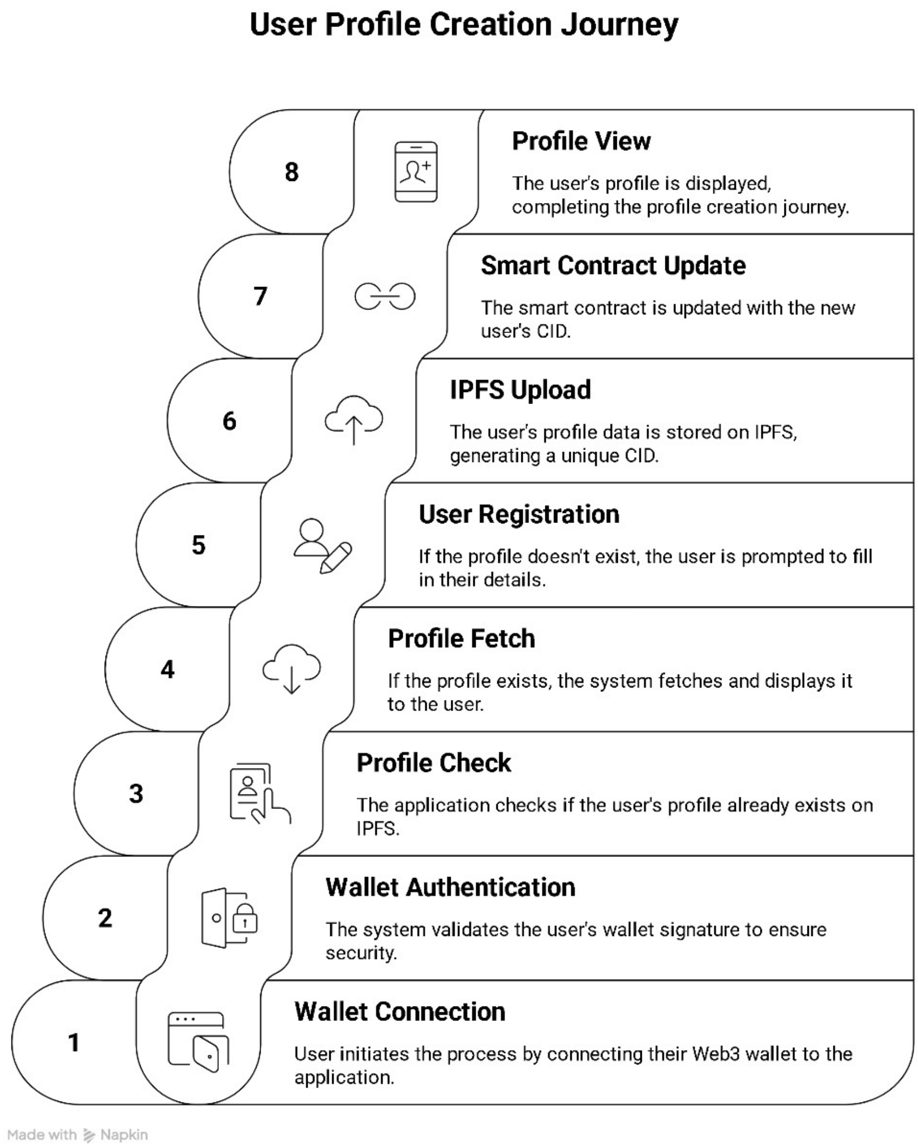

Figure 1: BlockRide System Overview and Architecture

4.3 Non-Functional Requirements

- Privacy/Data Minimization
- Integrity (immutability via CIDs)
- Availability (99.5% uptime for signaling)
- Latency ($\leq$ 3 seconds)
- Scalability
- Security (TLS, rate limits)
- Compliance (export/delete user data)

- Observability (logs, metrics)

4.4 Use Cases

- **UC-1** Authenticate with Wallet (Must Have)
- **UC-2** Register Profile (Must Have)
- **UC-3** Offer a Ride (Must Have)
- **UC-4** Request a Ride (Must Have)
- **UC-5** Confirm Ride (Must Have)
- **UC-6** View Profile (Useful)
- **UC-7** Logout (Useful)
- **UC-8** Ratings and Feedback (Optional)

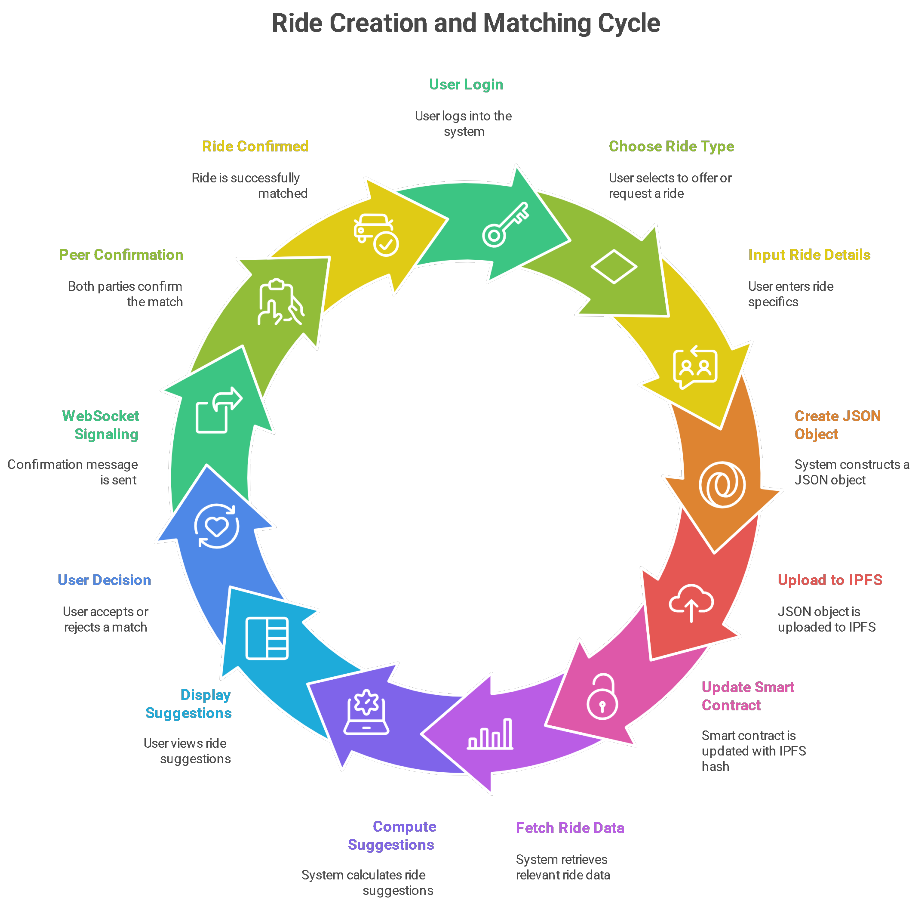

Figure 2: Ride Creation and Matching Process Flow

4.5 Acceptance Tests

- **AT-1**: Wallet Authentication within 2s
- **AT-2**: Profile Registration within 5s
- **AT-3**: Driver Offer submission confirmation within 3s
- **AT-4**: Rider Request suggestions within 3s
- **AT-5**: Real-Time Confirmation within 2s

4.6 User Interface Requirements

- Sidebar: Receive a Ride, Share a Ride, Contact Us, About Us, My Profile, Logout
- Pages: Home, Web3 Login, Signup, Ride Receiver, Ride Sharer, Profile
- Map Interaction: source/destination via map
- Loading & Error: overlays with actionable messages

4.7 Data Exchange Formats

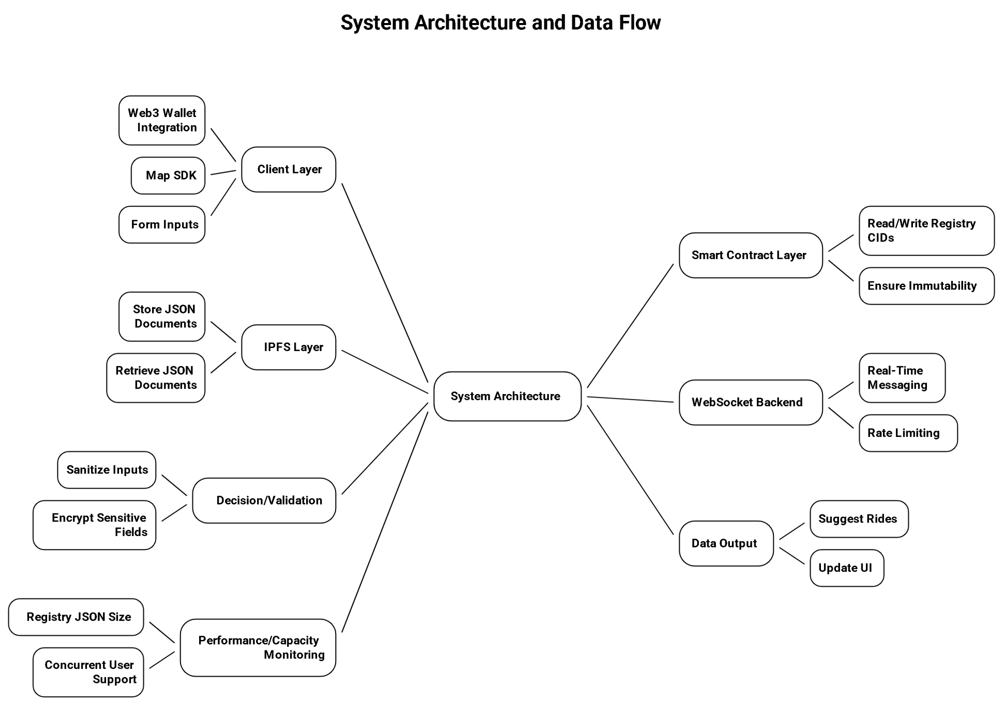

Figure 3: System Architecture and Data Flow Diagram

Smart Contract Interface:

- setRootUserHash(string cid)

- getRootUserHash()
- setRootRiderRequestHash(string cid)
- getRootRiderRequestHash()
- setRootDriverOfferHash(string cid)
- getRootDriverOfferHash()

JSON Examples:

Listing 1: User Registry JSON

```
{
  "0xUserAddress": "abc...profileCid",
  "0xAnotherAddress": "abc...profileCid2"
}
```

Listing 2: Profile JSON

```
{
  "username": "Aryan",
  "phone": "20055 00101",
  "city": "Delhi"
}
```

Listing 3: Rider Requests Registry

```
{
  "abc...riderProfileCid": {
    "source": { "latitude": 29.867583, "longitude": 77.894989 },
    "destination": { "latitude": 29.950000, "longitude": 77.980000 }
  }
}
```

Listing 4: Driver Offers Registry

```
{
  "pqr...driverProfileCid": {
    "car_number": "DL7 WW 1234",
    "car_model": "Merc Benz",
    "car_seats": 4,
    "free_car_seats": 2,
    "fare_per_km": 120,
    "source": { "latitude": 29.86, "longitude": 77.89 },
    "destination": { "latitude": 29.95, "longitude": 77.98 }
  }
}
```

Listing 5: WebSocket Signaling Messages

```
{
  "rider_name": "FGH",
```

```
  "rider_phone": "20255 50101",
  "source": { "latitude": 29.86, "longitude": 77.89 },
  "destination": { "latitude": 29.95, "longitude": 77.98 },
  "to_ipfs_hash": "abc..driverProfileCid",
  "rider_ipfs_hash": "fgh...riderProfileCid"
}
```

5 External Dependencies and Environment

- Client: Modern browser with Web3 wallet
- Blockchain: EVM compatible mainnet
- Storage: IPFS gateway
- Backend: WebSocket server
- Maps: Web map SDK

6 Constraints and Assumptions

- Payments, KYC, insurance out of MVP scope
- IPFS storage is immutable and public
- Matching is advisory; confirmation peer-to-peer

7 Security and Privacy

- Encrypt phone numbers
- Validate inputs
- Rate/limit write operations

8 Performance and Capacity

- Registry JSON capped to ~1MB
- Real-time signaling for ≤100 concurrent users (pilot)

9 References

- EVM and Web3 wallet interaction guidelines
- IPFS content addressing and gateway usage
- WebSocket protocol best practices
- Geospatial distance heuristics for matching