



RAIL.AI: Innovating and Automating for Efficiency, Safety, Experience

Track 2- Smart Rail Solution

Team Name: RAIL.AI

Team members (2-4):

- Dhruv Yadav
- Lakshya Raj Vijay
- Sai Sumedh

What is your IDEA?

RAIL.AI is a comprehensive solution ,designed to tackle three primary challenges in the rail sector: inefficient scheduling, lack of innovation, and data security concerns. It aligns with the Smart Rail Solutions track by focusing on smart scheduling, enhancing passenger experiences, and ensuring data security.

Addressing the Challenges and Alignment with Smart Rail Solutions track by:

- **Inefficient Scheduling:** Implements smart scheduling algorithms, inspired by **genetic algorithms**, to optimize train routing, reduce travel times, and minimize delays, enhancing system efficiency.
- **Lack of Innovation:** Introduces **dynamic pricing**, **crowdsourced safety reporting**, **personalized travel recommendations**, and a **collaborative platform for lost & found** items to improve passenger experiences.
- **Data Security Concerns:** Utilizes **blockchain** technology for secure data storage and access control, protecting railway information from unauthorized access.

In essence, RAIL.AI embodies the soul of the Smart Rail Solutions track by leveraging technology to enhance efficiency, innovation, and security within the rail sector.

We have implemented 70% of the complete solution in the 30hrs of hackathon.

Why & how is your idea DIFFERENT?

Inefficient scheduling -

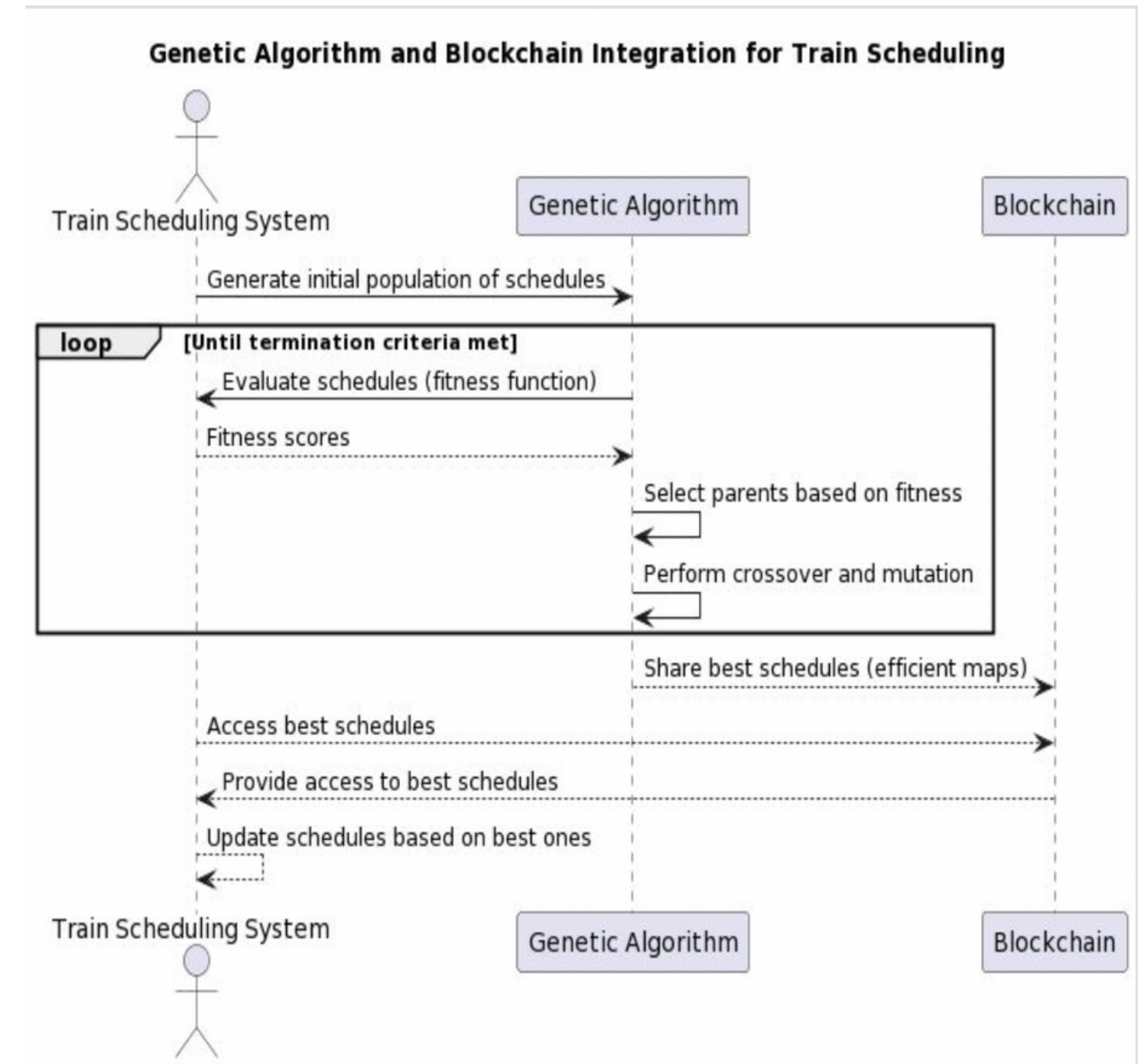
The genetic algorithm optimizes train schedules by generating, evaluating, and iteratively refining potential timetables, selecting the most efficient ones based on key performance metrics. Blockchain technology records the optimal schedules, ensuring consistency and transparency across the rail network.

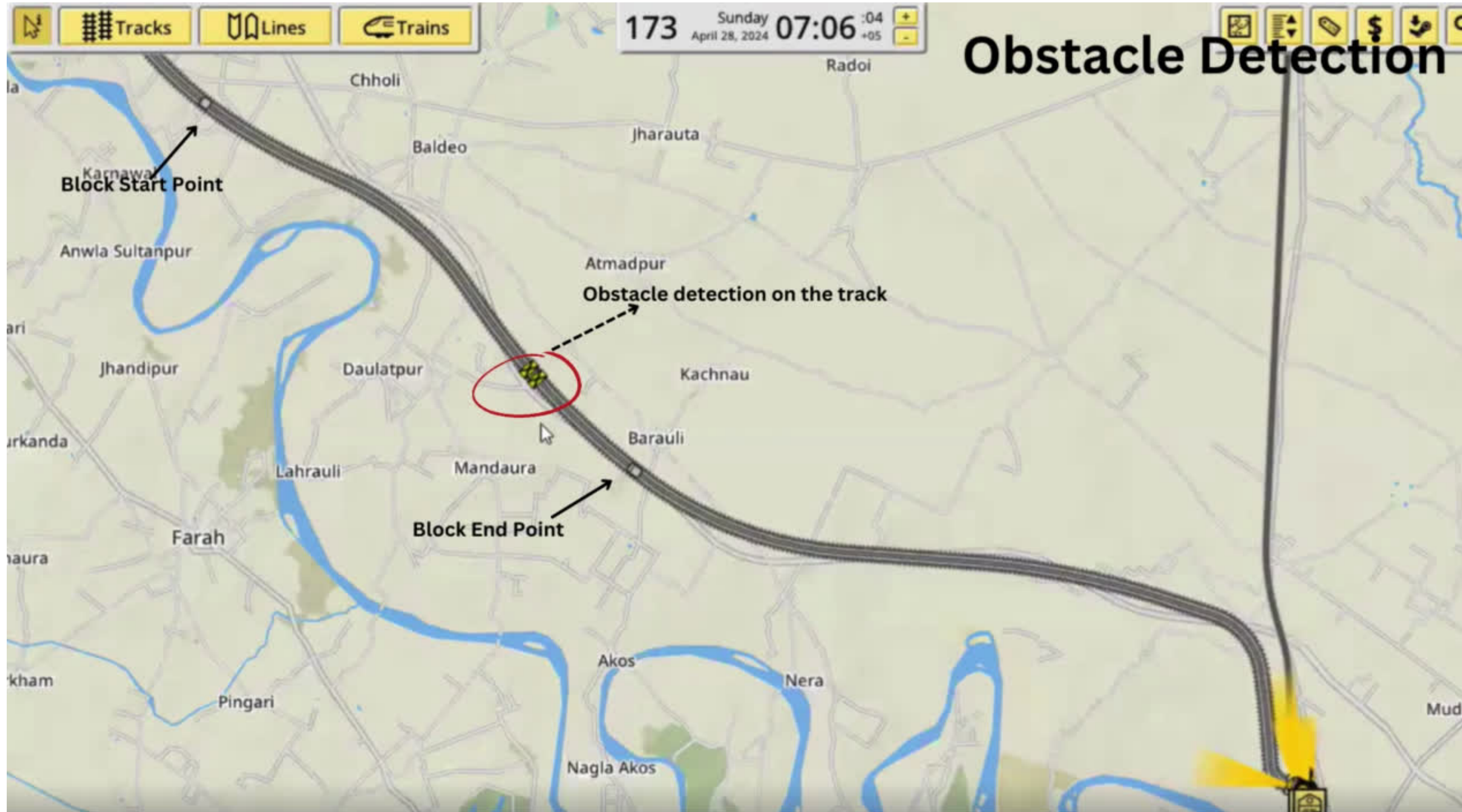
Future Potential-

Real-Time Adaptive Scheduling: The system could evolve to adapt schedules in real-time based on current conditions, such as delays, weather, and passenger demand.

Business Use Case-

- Enhanced Efficiency:** Optimized train routing and scheduling lead to reduced travel times as currently, In a span of 7 months, Indian Railways experienced delays equivalent to 24 years, with passenger and express trains losing around 55 lakh and 71 lakh minutes, respectively.
- Resource Optimization:** Better scheduling allows for optimal use of trains, crews and maintenance, reducing operational costs by 61%.
- Safety Improvements:** By avoiding scheduling conflicts, the risk of accidents and service disruptions is lowered.





Why & how is your idea DIFFERENT?

Dynamic Pricing and Demand Prediction

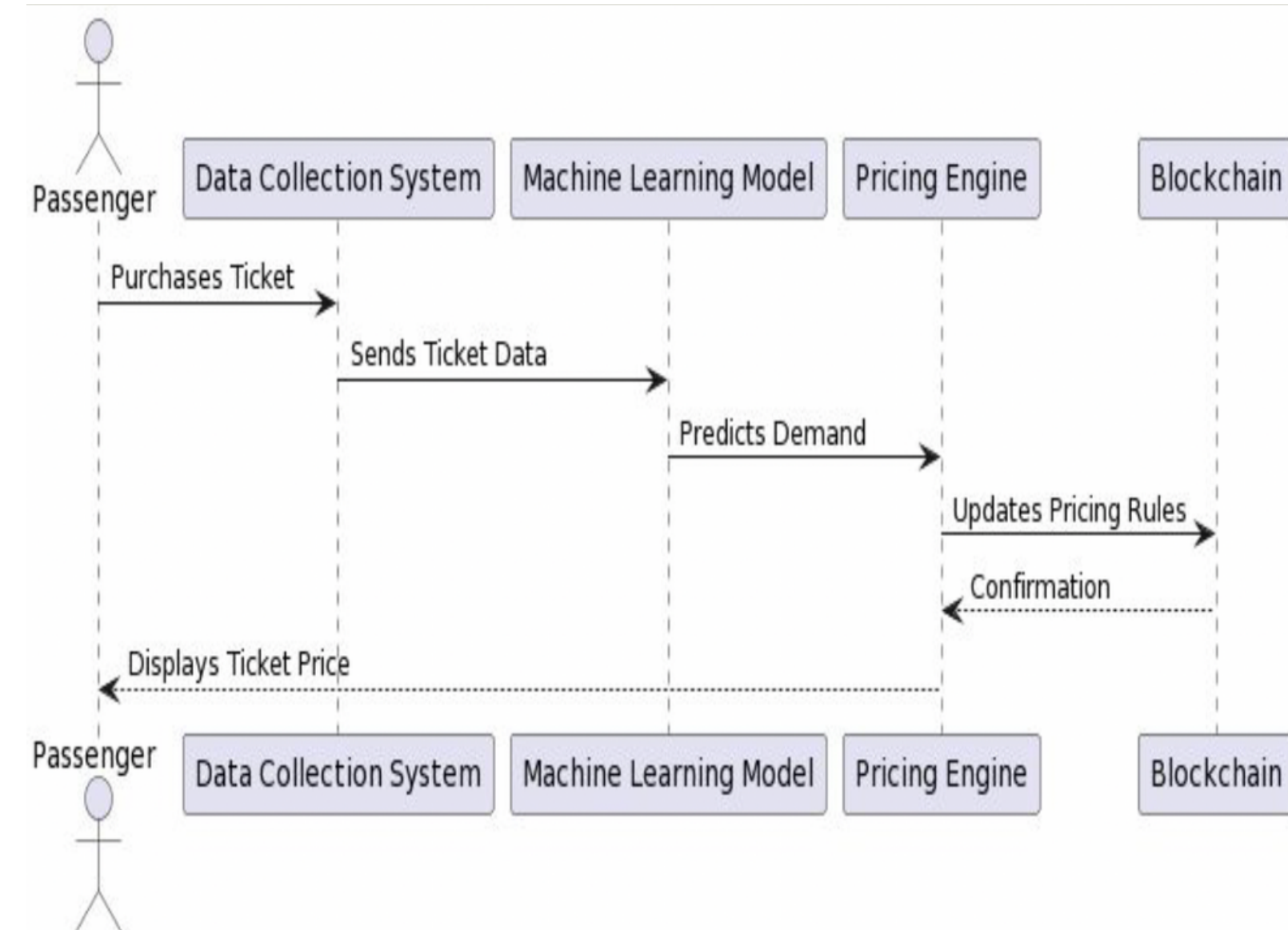
The system employs AI algorithms to analyze historical data, current booking trends, and external factors such as holidays or events to accurately predict real-time demand for train routes. Leveraging this data, dynamic pricing strategies are implemented, adjusting ticket prices based on demand, seat availability, and travel times. This approach aims to maximize revenue while incentivizing passengers to travel on less crowded trains, thus balancing load across the network.

Future Potential-

1. **Personalized Pricing:** Incorporate passenger preferences and loyalty data to offer personalized pricing options.
2. **Predictive Maintenance:** Incorporating maintenance data could help adjust prices based on vehicle availability, optimizing operations and reducing downtime.

Business Use Case-

1. **Revenue Optimization:** Real-time pricing adjustments based on demand can maximize revenue and improve vehicle utilization.
2. **Targeted Marketing:** Customer segmentation enables personalized offers, especially for frequent travelers or loyalty members.
3. **Load Management:** Incentivizing off-peak travel through pricing can distribute passenger load more evenly, enhancing the travel experience.



Why & how is your idea DIFFERENT?

Crowdsourced Safety Reporting-

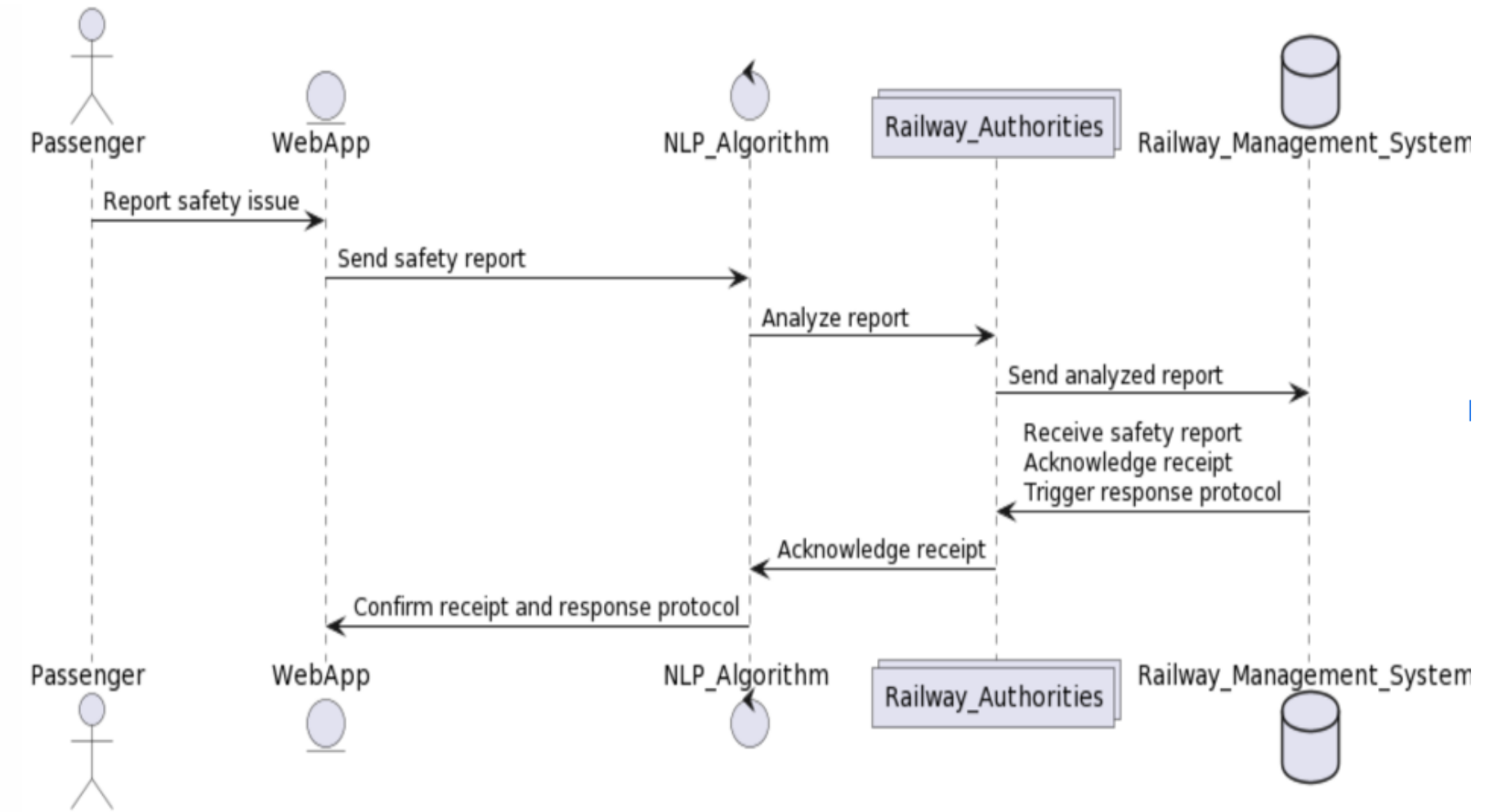
1. **Real-time Safety Reporting:** Enable passengers to report safety hazards or incidents instantly through a chatbot.
2. **Emergency Communication:** Provide features for emergency notifications and direct communication with railway authorities.

Future Potential-

1. **Predictive Safety Analytics:** Analyze reported incidents to identify patterns and predict potential safety issues.
2. **Integration with IoT Sensors:** Integrate with IoT sensors on trains for automated incident detection.

Business Use Case-

1. **Predictive Maintenance:** By identifying equipment issues before they lead to failures, predictive maintenance minimizes disruptions and extends the lifespan of railway assets, leading to cost savings.
2. **Incident Prevention:** Immediate reporting and addressing of safety hazards prevent accidents, which can be costly in terms of damages and potential legal liabilities.
3. **Resource Optimization:** Real-time data allows for better planning and allocation of maintenance resources, avoiding unnecessary work and focusing on areas that require attention.



Why & how is your idea DIFFERENT?

Personalized Travel Recommendations-

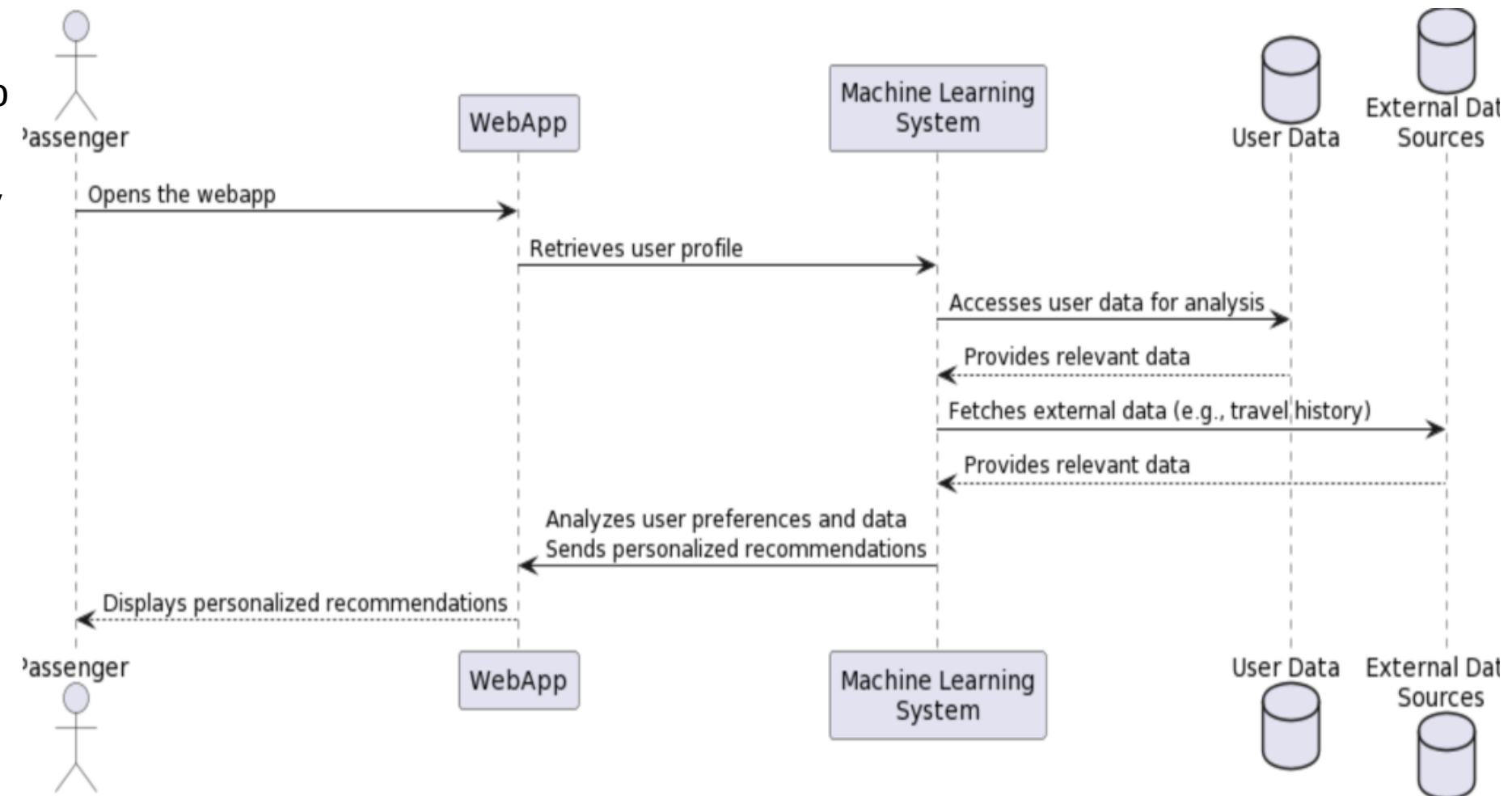
1. **Personalized Recommendations:** Offer travel recommendations tailored to individual preferences, travel history, and current location.
2. **Route Optimization:** Suggest optimal routes, train connections, and nearby attractions to enhance the overall travel experience.

Future Potential-

1. **Eco-friendly Route Recommendations:** Recommend routes with lower energy consumption for specific journeys, promoting sustainable travel choices and optimizing overall railway network efficiency.
2. **Real-time Adjustments Based on External Factors:** Incorporate real-time adjustments based on various external factors such as weather conditions, events, and infrastructure disruptions.

Business Use Case-

1. **Route Optimization and Upselling:** Recommend faster routes with higher fares or scenic routes with supplementary services like on-board meals to increase revenue per passenger.
2. **Personalized Pass Promotions:** Recommend season passes or discounted tickets based on travel frequency and patterns, encouraging increased railway usage.



Why & how is your idea DIFFERENT?

Collaborative Lost and Found Platform

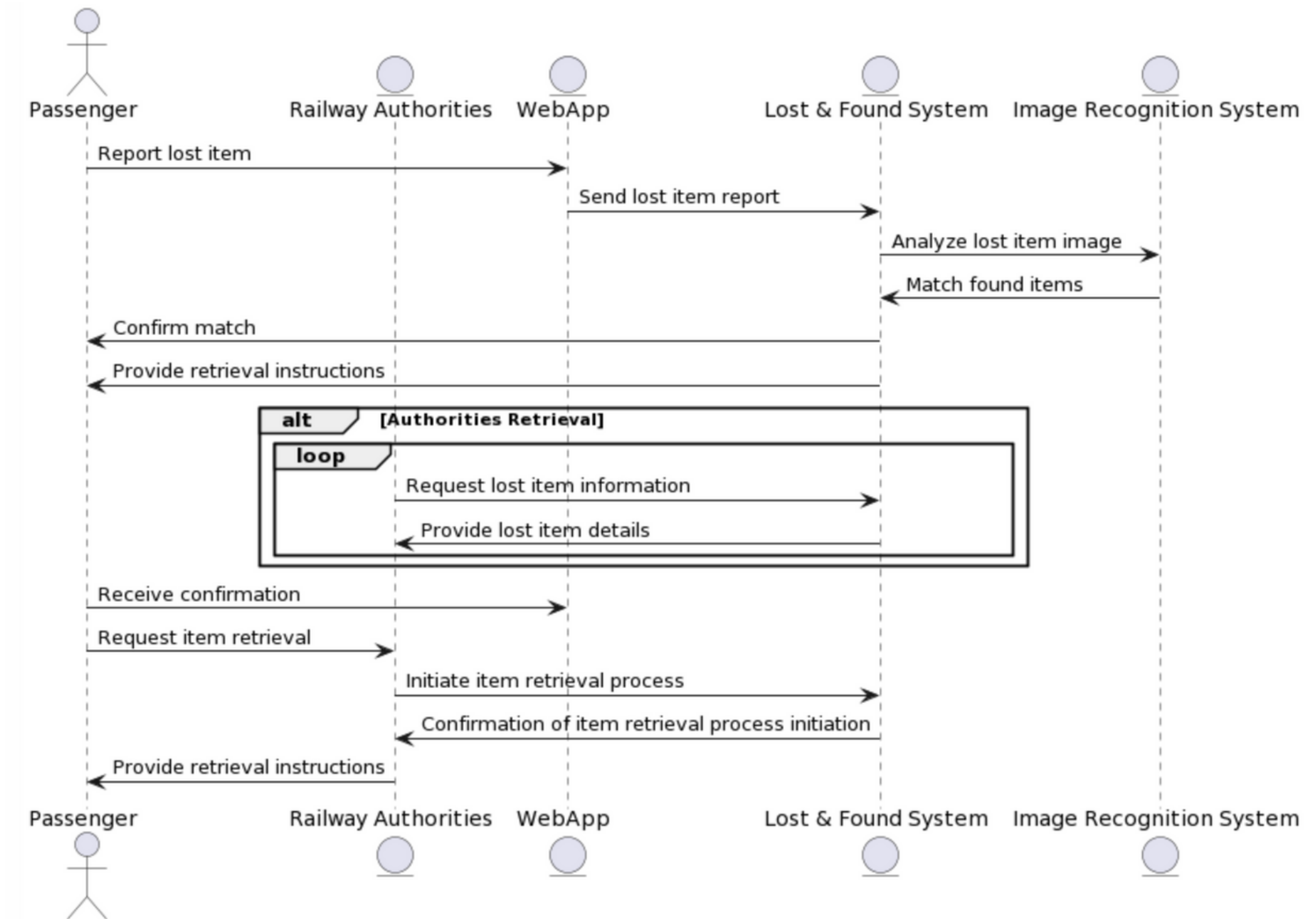
1. **Lost Item Reporting:** Allow passengers to report lost items and browse found items across stations.
2. **AI-Powered Matching:** Utilize image recognition and AI to match lost items with descriptions for expedited retrieval.

Future Potential-

1. **Blockchain Integration:** Implement blockchain technology for secure and transparent item tracking.
2. **Integration with Smart Lockers:** Partner with smart locker providers for automated item storage and retrieval.

Business Use Case-

1. **Lost Item Trend Analysis:** Analyze data on frequently lost items to identify potential security vulnerabilities in specific stations, allowing for targeted security measures.
2. **Automated Lost and Found Process:** Reduce manual paperwork and streamline the reporting and retrieval process, freeing up staff time for other critical tasks.



Why & how is your idea DIFFERENT?

Eco-friendly Train Infrastructure

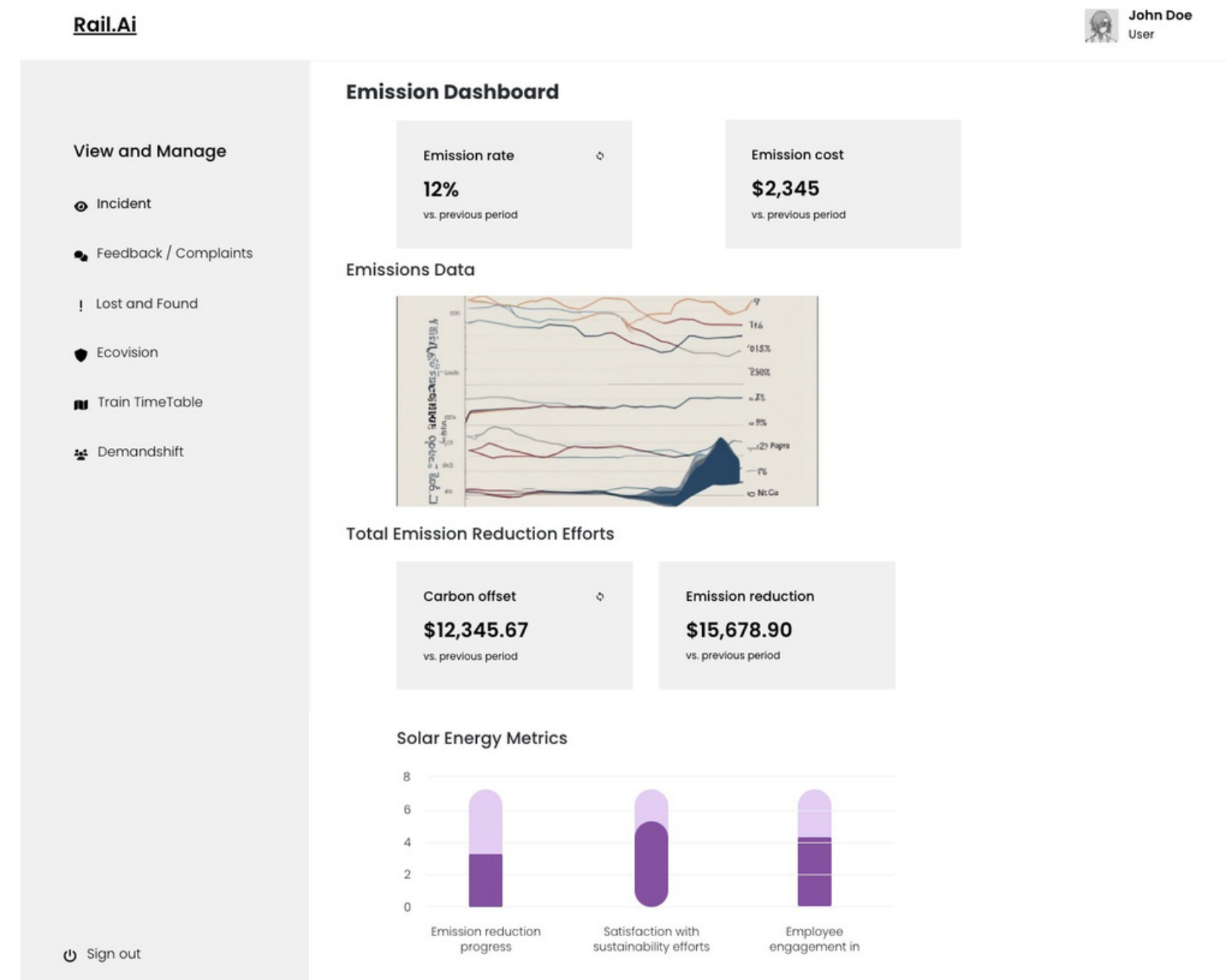
1. **Eco-friendly Features:** Introduce trains with solar panels energy-efficient lighting, and regenerative braking systems to reduce environmental impact.
2. **Monitoring Dashboard:** Develop dashboards for monitoring train emissions and energy usage in real-time.

Future Potential-

- **Hybrid Train Technology:** Hybrid trains using electric and hydrogen power reduce emissions, providing a clean alternative in non-electrified areas.
- **Carbon Offsetting Programs:** Railways can neutralize emissions by investing in carbon reduction projects like reforestation and renewable energy.

Business Use Case-

1. **Access to Incentives:** Enables companies to benefit from government eco-incentives.
2. **Increased Operational Efficiency:** Reduces energy consumption and operational costs.
3. **Building Community Relations:** Strengthens ties with local communities through sustainability efforts.





Rail.AI
Admin Access

Admin Log in

Username

Password

A captcha image showing the text "Z5M3M0" in a distorted, pixelated font.

Log in

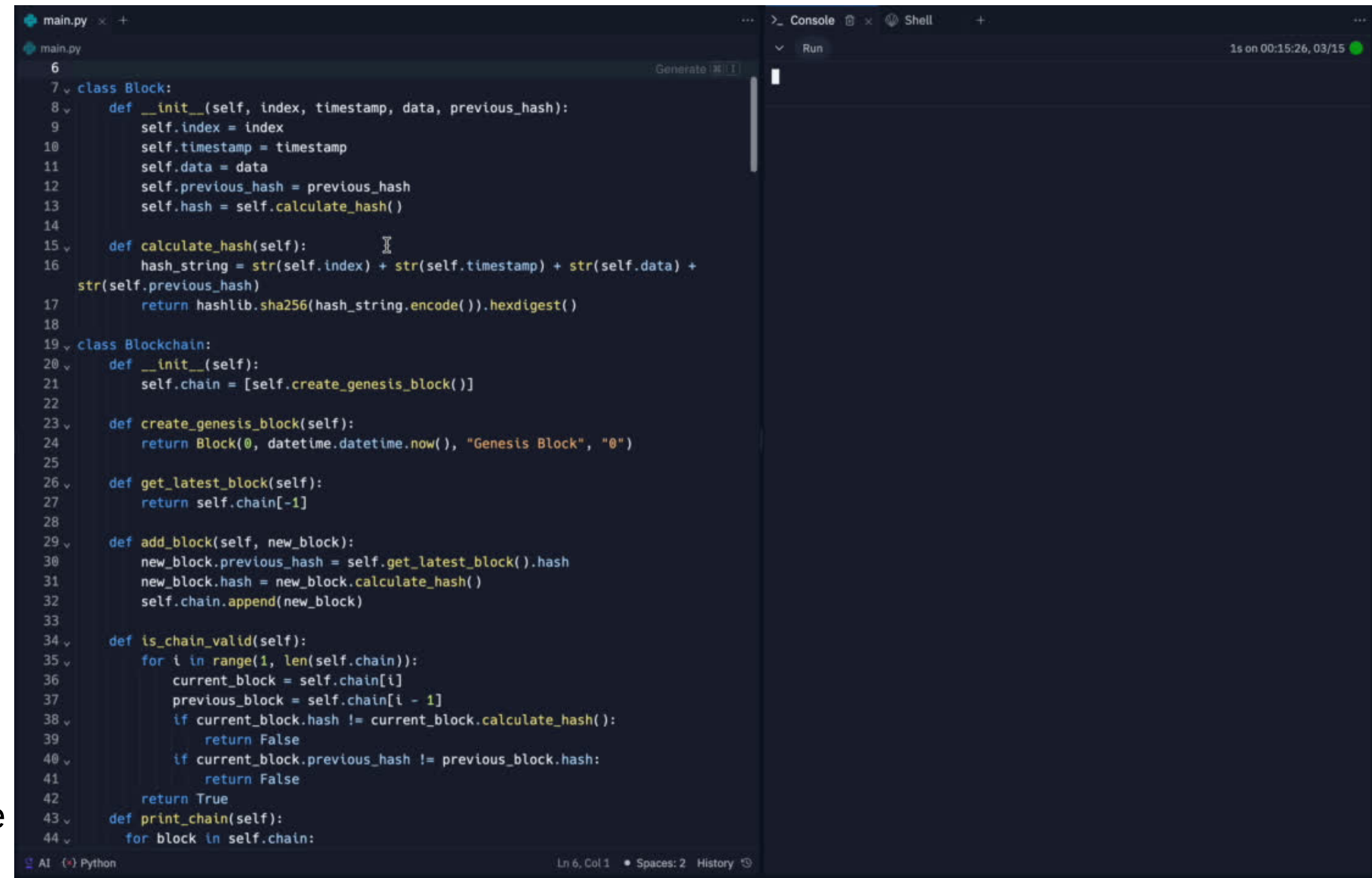
[Forgot your password?](#)

Why & how is your idea DIFFERENT?

Data Security Concerns

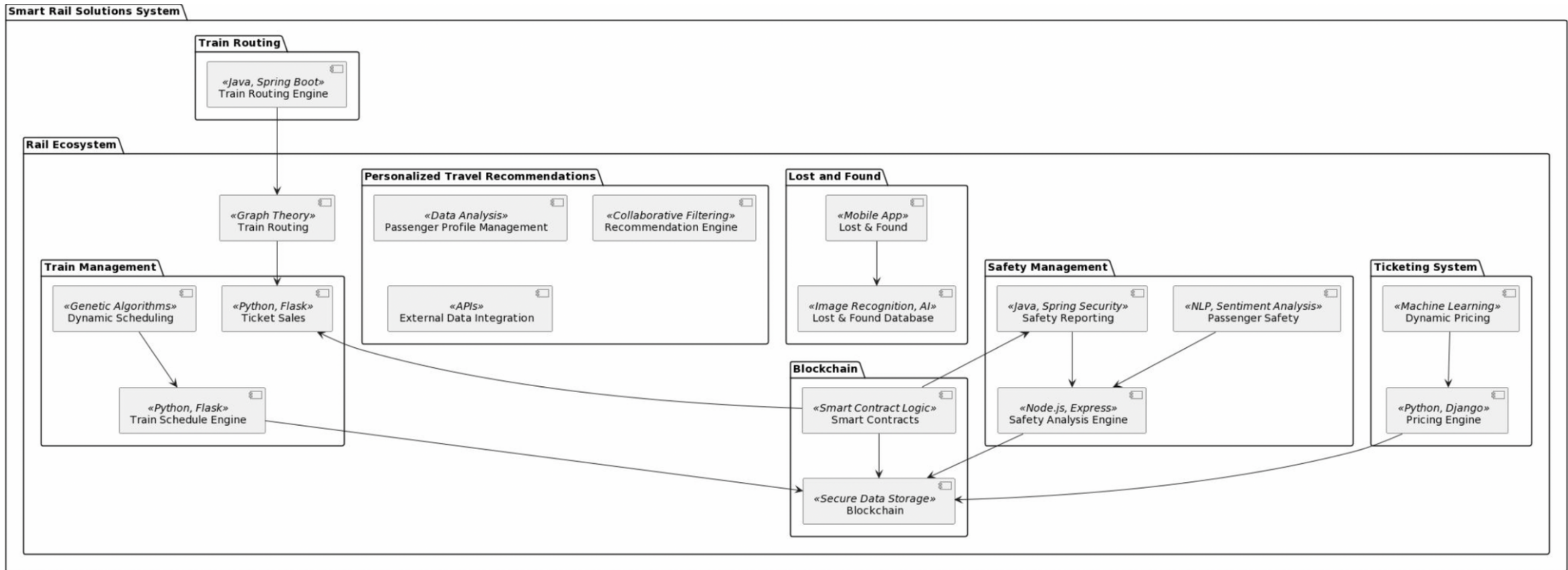
Blockchain technology enhances railway data security through:

1. **Immutable Storage:** Ensures recorded data cannot be changed, maintaining data integrity.
2. **Transparency:** Allows authorized parties to audit transactional data in real-time.
3. **Decentralization:** Reduces single points of failure by distributing data across multiple nodes.
4. **Secure Access:** Restricts data interaction to users with private keys.
5. **Consensus Validation:** Employs mechanisms like PoW or PoS to verify transactions before recording.
6. **Encryption:** Protects sensitive information from unauthorized access.
7. **Algorithm Integration:** Combines with algorithms like genetic algorithm to optimize train scheduling and security.

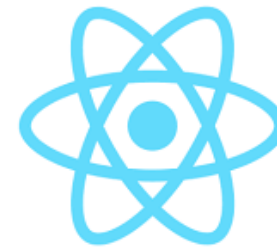
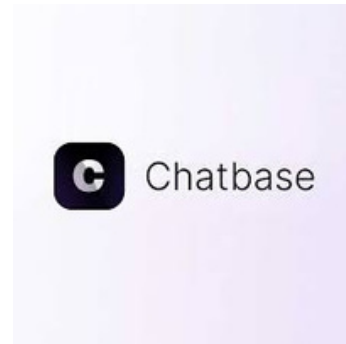


```
main.py x +
main.py
6
7 class Block:
8     def __init__(self, index, timestamp, data, previous_hash):
9         self.index = index
10        self.timestamp = timestamp
11        self.data = data
12        self.previous_hash = previous_hash
13        self.hash = self.calculate_hash()
14
15    def calculate_hash(self):
16        hash_string = str(self.index) + str(self.timestamp) + str(self.data) +
17        str(self.previous_hash)
18        return hashlib.sha256(hash_string.encode()).hexdigest()
19
20 class Blockchain:
21     def __init__(self):
22         self.chain = [self.create_genesis_block()]
23
24     def create_genesis_block(self):
25         return Block(0, datetime.datetime.now(), "Genesis Block", "0")
26
27     def get_latest_block(self):
28         return self.chain[-1]
29
30     def add_block(self, new_block):
31         new_block.previous_hash = self.get_latest_block().hash
32         new_block.hash = new_block.calculate_hash()
33         self.chain.append(new_block)
34
35     def is_chain_valid(self):
36         for i in range(1, len(self.chain)):
37             current_block = self.chain[i]
38             previous_block = self.chain[i - 1]
39             if current_block.hash != current_block.calculate_hash():
40                 return False
41             if current_block.previous_hash != previous_block.hash:
42                 return False
43         return True
44
45     def print_chain(self):
46         for block in self.chain:
```


Solution Overview



Technologies





THANK YOU

GlobalLogic®
A Hitachi Group Company

30Hacks
Coding and Caffeine is all you need!

Supported by **Hitachi Group**