

The background features a dark blue gradient with three glowing, translucent 3D torus shapes. One large ring is positioned at the top left, another smaller one is at the bottom left, and a third smaller one is located on the right side.

Vehicle Routing System

Group 8

Members

**Nguyen Van
Quoc**

(Team Leader)

- Project and Progress Management
- MRA Development
- Optimize algorithm implementation (GA, ACO, and PSO)
- Team Coordination
- Code Review

**Tran Hung
Quoc Tuan**

(Frontend
Developer)

- User interface development
- Dashboard design
- Route visualization
- User interaction

**Nguyen Ha
Minh Chau**

(QA and
Documentation)

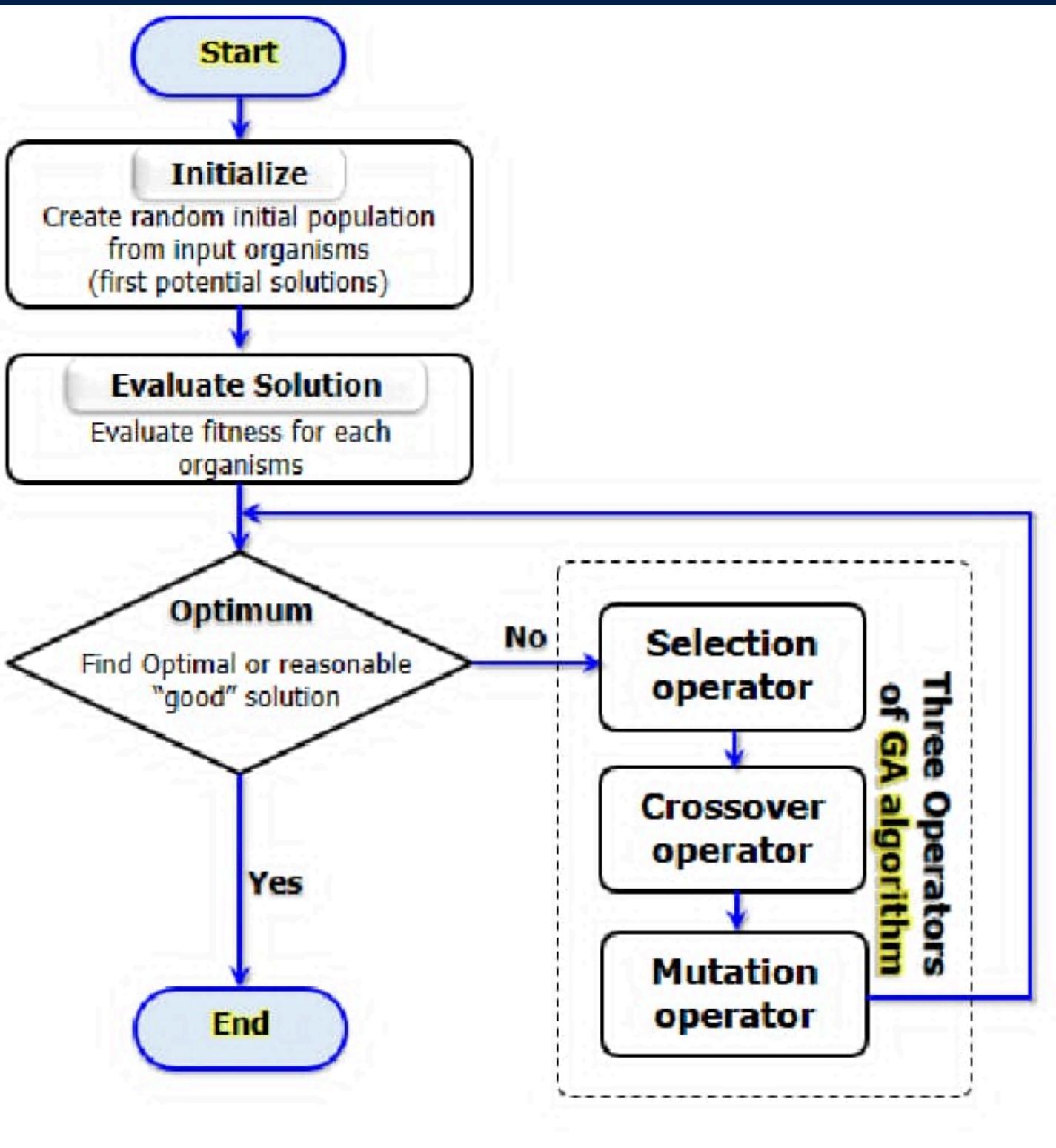
- Write test cases
- Create documentation
- Configuration management
- Support other team members

**Mai Hoang
Dai Vy**

(Backend
Developer)

- Delivery Agents (DAs) development
- Agent communication protocol implementation
- Business logic handling

GA workflow:



Dataset for DAs:

A	B	C	D	E	F	G	H	I	J	K
Order_ID	Material_ID	Item_ID	Source	Destination	Available_Tim	Deadline	Danger_Type	Area	Weight	
1	A140109	B-6128	P01-79c46a0	City_61	City_54	#####	#####	type_1	38880	30920000
2	A140109	B-6128	P01-43f08b0f	City_61	City_54	#####	#####	type_1	38880	30920000
3	A140109	B-6128	P01-899d738	City_61	City_54	#####	#####	type_1	38880	30920000
4	A140109	B-6128	P01-acc23cd	City_61	City_54	#####	#####	type_1	38880	30920000
5	A140109	B-6128	P01-cd0377d	City_61	City_54	#####	#####	type_1	38880	30920000
6	A140109	B-6128	P01-ba00d24	City_61	City_54	#####	#####	type_1	38880	30920000
7	A140109	B-6128	P01-6994ea5	City_61	City_54	#####	#####	type_1	38880	30920000
8	A140109	B-6128	P01-d06ba80	City_61	City_54	#####	#####	type_1	38880	30920000
9	A140109	B-6128	P01-06c3fc32	City_61	City_54	#####	#####	type_1	38880	30920000
10	A140109	B-6128	P01-69f2a2b9	City_61	City_54	#####	#####	type_1	38880	30920000
11	A140110	B-6128	P01-1fc56a04	City_61	City_54	#####	#####	type_1	38880	30920000
12	A140110	B-6128	P01-ce8e41e	City_61	City_54	#####	#####	type_1	38880	30920000
13	A140110	B-6128	P01-e0ff5073	City_61	City_54	#####	#####	type_1	38880	30920000
14	A140110	B-6128	P01-0cd66c2	City_61	City_54	#####	#####	type_1	38880	30920000
15	A140110	B-6128	P01-afb96755	City_61	City_54	#####	#####	type_1	38880	30920000
16	A140110	B-6128	P01-9901aba	City_61	City_54	#####	#####	type_1	38880	30920000
17	A140110	B-6128	P01-32ea426	City_61	City_54	#####	#####	type_1	38880	30920000
18	A140110	B-6128	P01-8819dc2	City_61	City_54	#####	#####	type_1	38880	30920000
19	A140112	B-6128	P01-9046e25	City_61	City_54	#####	#####	type_1	38880	30920000
20	A140112	B-6128	P01-da3626c	City_61	City_54	#####	#####	type_1	38880	30920000
21	A140112	B-6128	P01-6b71ea7	City_61	City_54	#####	#####	type_1	38880	30920000
22	A140112	B-6128	P01-84ac394	City_61	City_54	#####	#####	type_1	38880	30920000
23	A140112	B-6128	P01-17b9602	City_61	City_54	#####	#####	type_1	38880	30920000
24	A140112	B-6128	P01-17b9602	City_61	City_54	#####	#####	type_1	38880	30920000

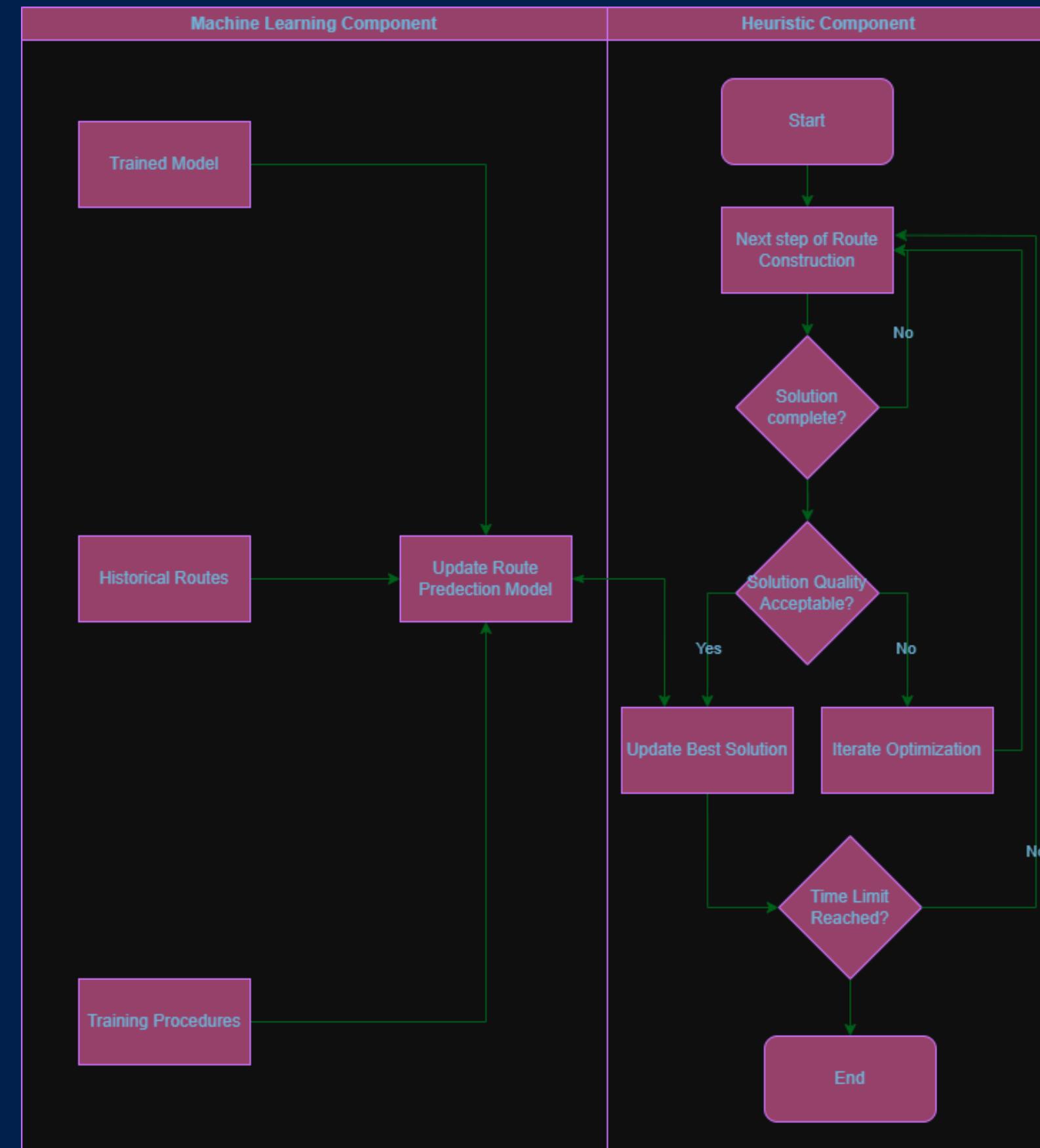
Distance Dataset:

	A	B	C	D
1	Source	Destination	Distance(M)	
2	City_24	City_47	1114251	
3	City_24	City_31	97187	
4	City_24	City_54	1716028	
5	City_24	City_53	1729925	
6	City_24	City_19	1594107	
7	City_24	City_12	774894	
8	City_24	City_46	1146028	
9	City_24	City_45	1147045	
10	City_24	City_51	1107377	
11	City_24	City_6	1688216	
12	City_24	City_20	1615472	
13	City_24	City_56	1636630	
14	City_24	City_35	1053903	
15	City_24	City_48	1300876	

Truck Specification

Truck Type (length in m)	Inner Size (m^2)	Weight Capacity (kg)	Cost Per KM	Speed (km/h)
16.5	16.1×2.5	10000	3	40
12.5	12.1×2.5	5000	2	40
9.6	9.1×2.3	2000	1	40

VRP (Vehicle Routing Problem) flowchart



```
class RouteOptimizerGA:  
>     def __init__(self, data_processor: DataProcessor): ...  
  
>     def optimize(self) -> List[Route]: ...  
  
>     def _initialize_population(self) -> List[List[Route]]: ...  
  
>     def _create_randomized_solution(self) -> List[Route]: ...  
  
>     def _build_route(self, truck_type: str, available_orders: List[str], randomize: bool = False) -> Optional[Route]: ...  
  
>     def _calculate_fitness(self, solution: List[Route]) -> float: ...  
  
>     def _selection(self, population: List[List[Route]], fitness_scores: List[float]) -> List[List[Route]]: ...  
  
>     def _crossover(self, parent1: List[Route], parent2: List[Route]) -> List[Route]: ...  
  
>     def _create_route_with_orders(self, truck_type: str, order_ids: List[str]) -> Optional[Route]: ...  
  
>     def _mutate(self, solution: List[Route]) -> List[Route]: ...  
  
>     def _swap_orders_mutation(self, solution: List[Route]) -> List[Route]: ...  
  
>     def _create_route_without_parcel(self, route: Route, parcel_id: int) -> Optional[Route]: ...  
  
>     def _add_order_to_route(self, route: Route, order_id: str, truck_type: str) -> Optional[Route]: ...  
  
>     def _change_truck_mutation(self, solution: List[Route]) -> List[Route]: ...  
  
>     def _reorder_stops_mutation(self, solution: List[Route]) -> List[Route]: ...  
  
>     def _repair_solution(self, solution: List[Route]) -> List[Route]: ...  
  
>     def _check_route_feasibility(self, route: Route, truck_type: str) -> Tuple[bool, str]: ...  
  
>     def _get_warehouse_location(self) -> Location: ...  
  
>     def _create_location(self, city_name: str) -> Location: ...  
  
>     def evaluate_solution(self, routes: List[Route]) -> Dict: ...
```

Route Optimizer flow using Genetic Algorithm

```
def main():
    """Test the GA route optimizer"""
    try:
        from src.data.data_processor import DataProcessor

        # Initialize data
        processor = DataProcessor()
        processor.load_data('distance.csv', 'order_large.csv')
        processor._create_distance_matrix()
        processor._process_time_windows()

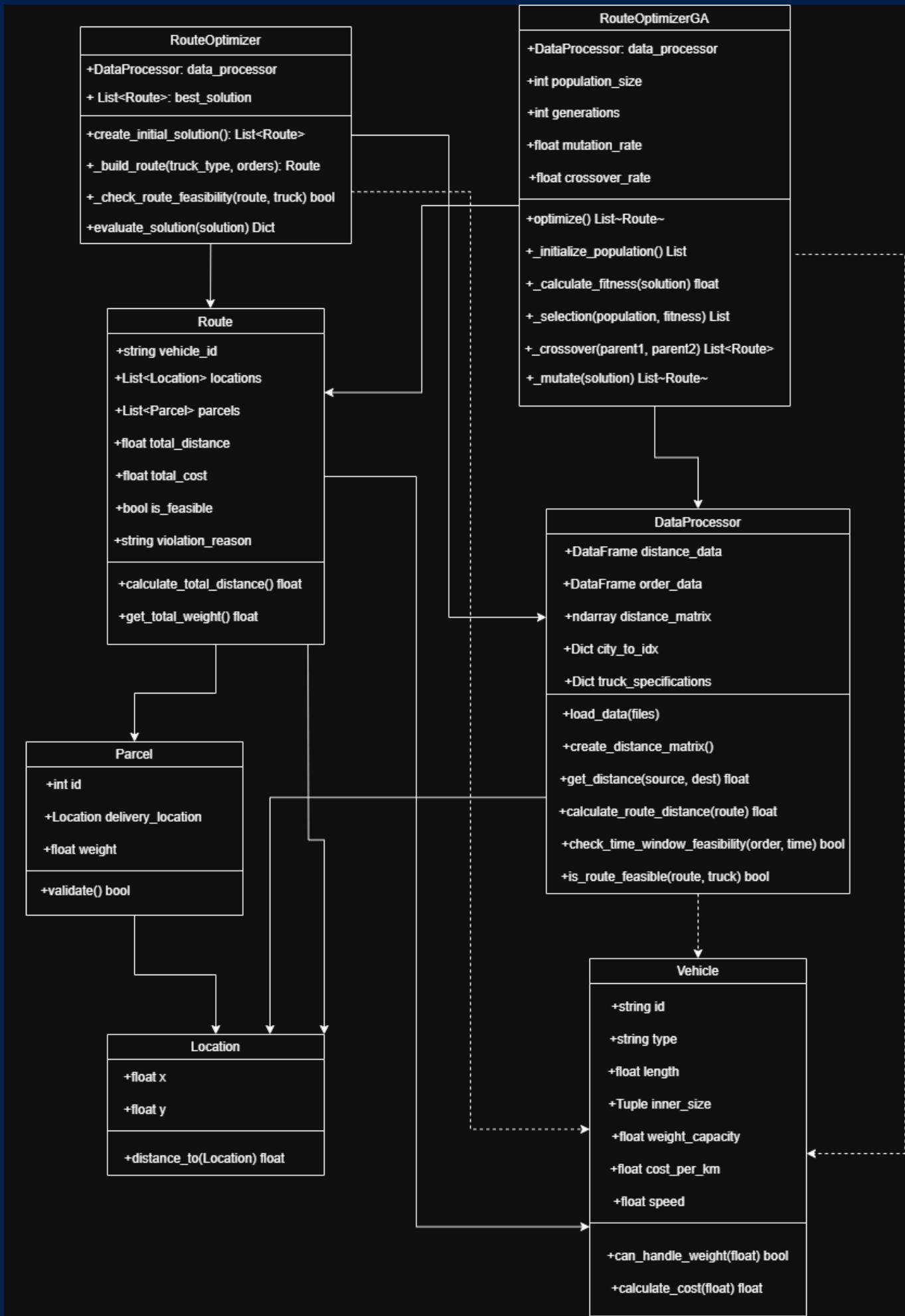
        # Create optimizer
        optimizer = RouteOptimizerGA(processor)

        # Run optimization
        solution = optimizer.optimize()

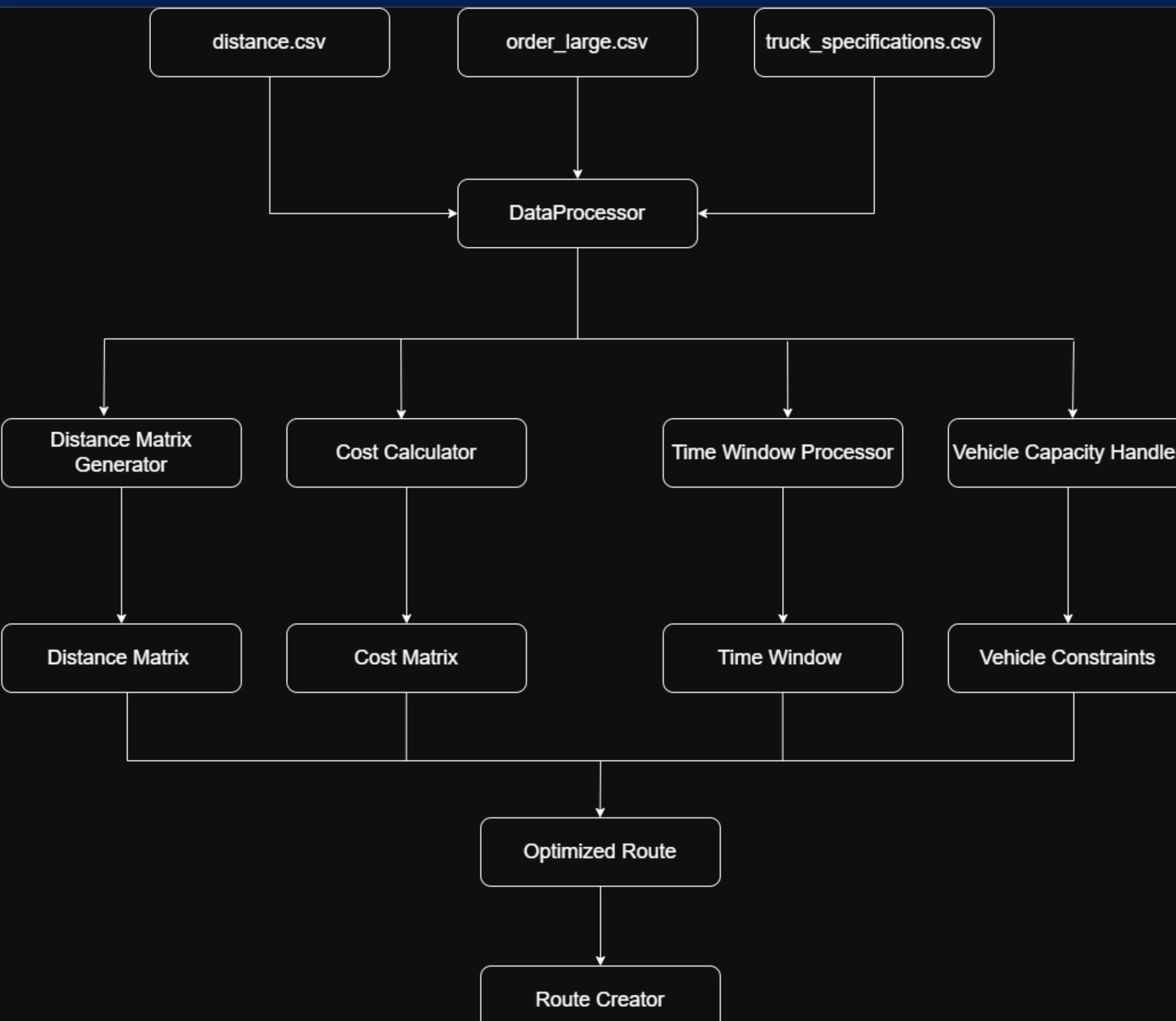
        # Print solution details
        if solution:
            print("\nBest Solution Details:")
            for i, route in enumerate(solution):
                print(f"\nRoute {i+1} ({route.vehicle_id}):")
                print(f"Parcels: {len(route.parcels)}")
                print(f"Distance: {route.total_distance:.2f} km")
                print(f"Cost: ${route.total_cost:.2f}")
                print(f"Load: {route.get_total_weight():.2f} kg")

        except Exception as e:
            print(f"Error in main: {e}")
            import traceback
            traceback.print_exc()
```

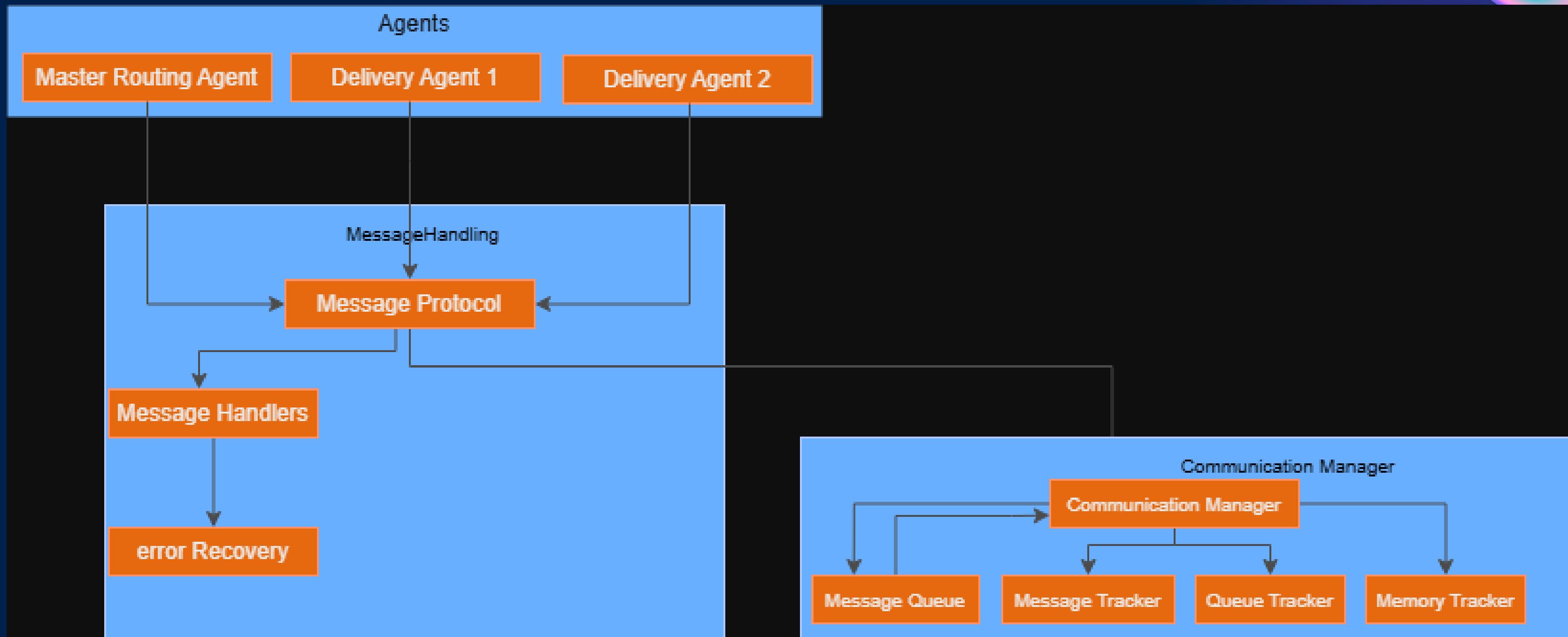
Data Class Architecture



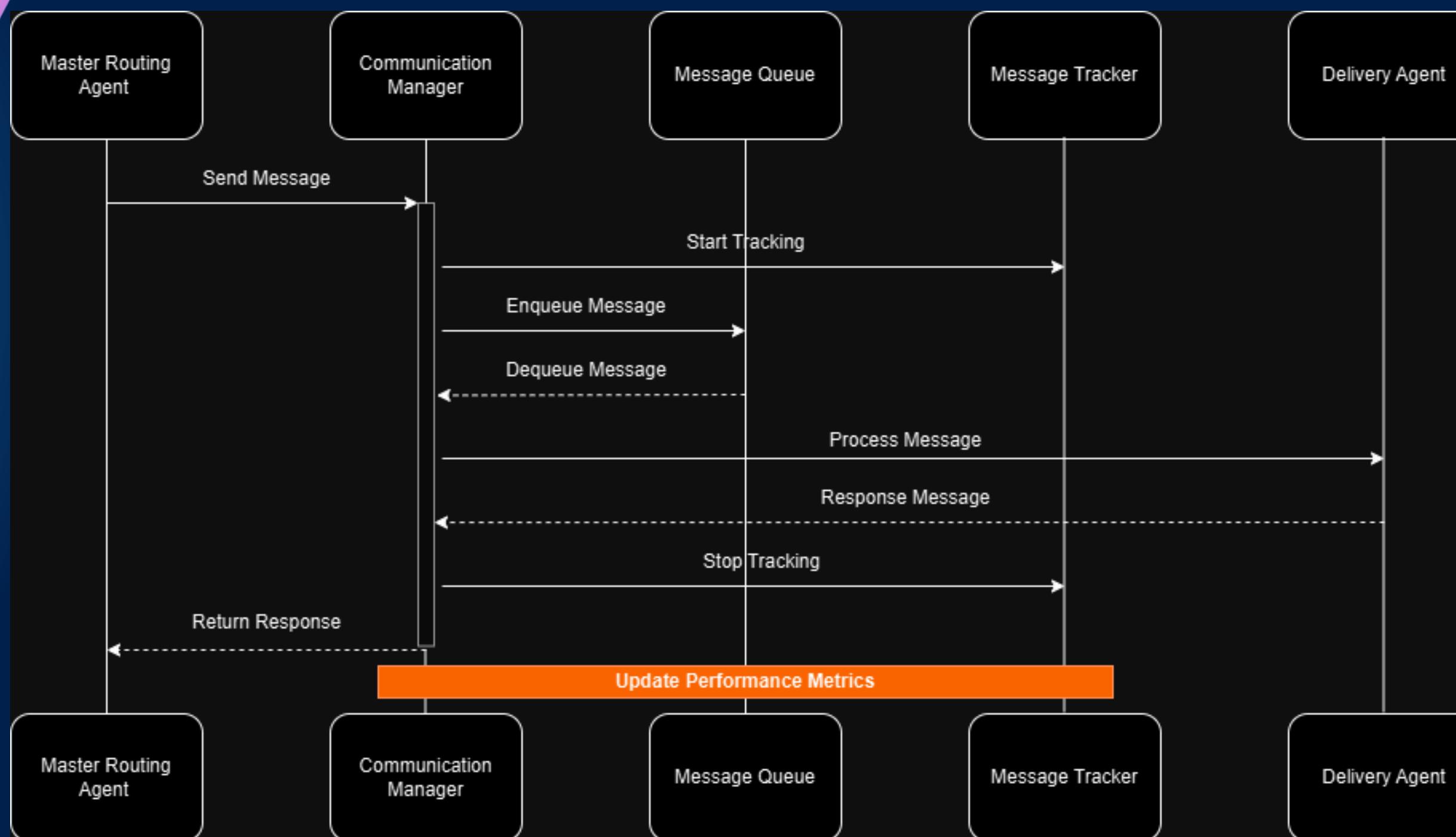
Data Processing Flow



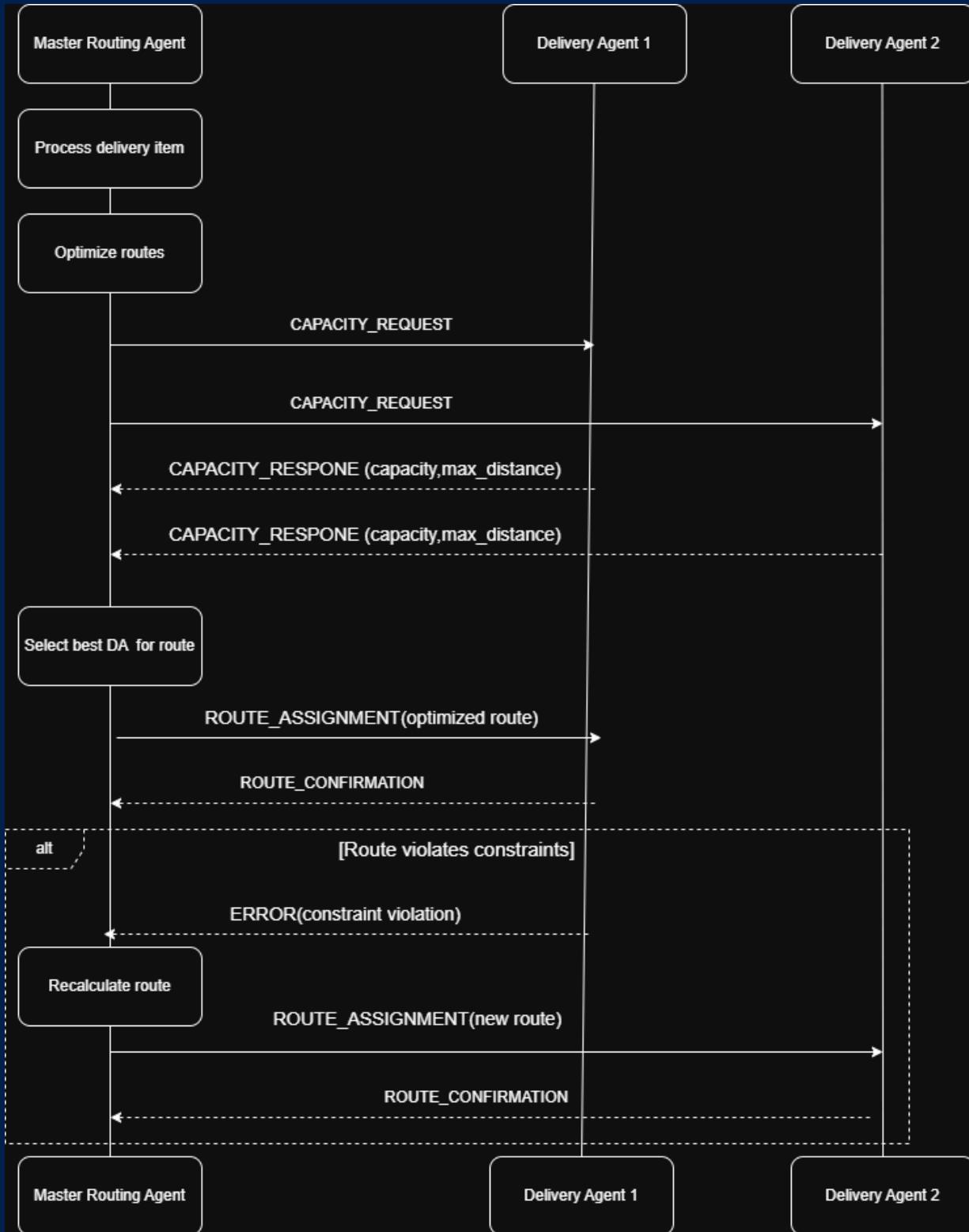
Communication Architecture



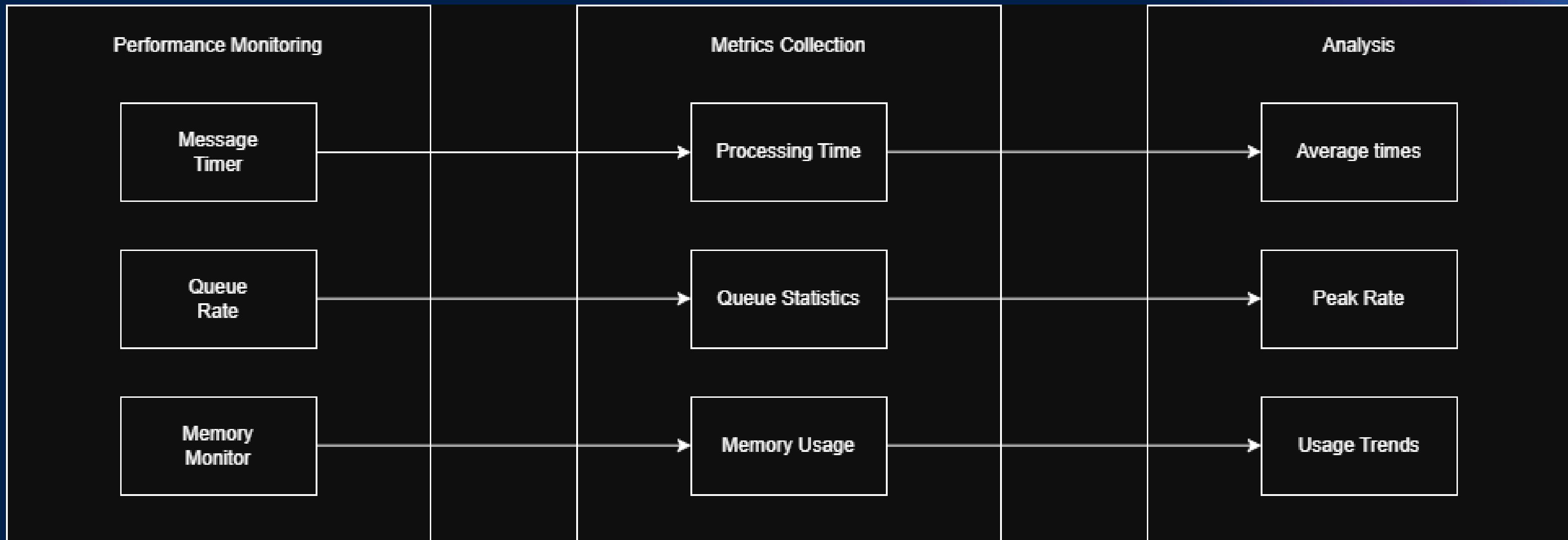
Message Processing Diagram



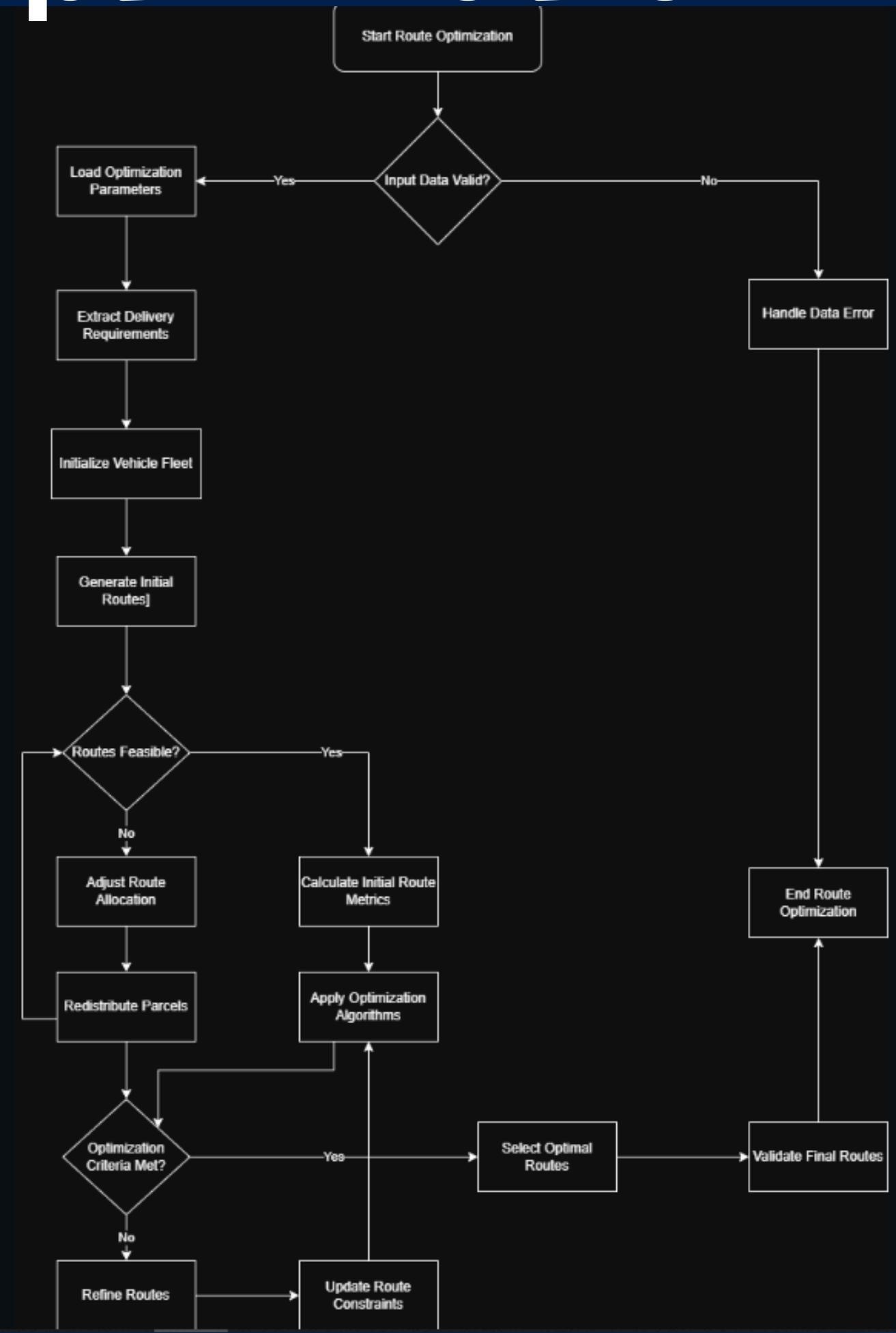
System Processing Diagram



Performance Monitoring Indicator



Route Optimization Flowchart



Timeline

W2

- Project Setup and Planning

W3

- Basic Infrastructure

W4

- Agent Communication

W5 – 6

- Data Processing & Initial Optimization Setup.
- Core Optimization Algorithm Implementation.

W7

- Advanced Route Optimization Features

W8

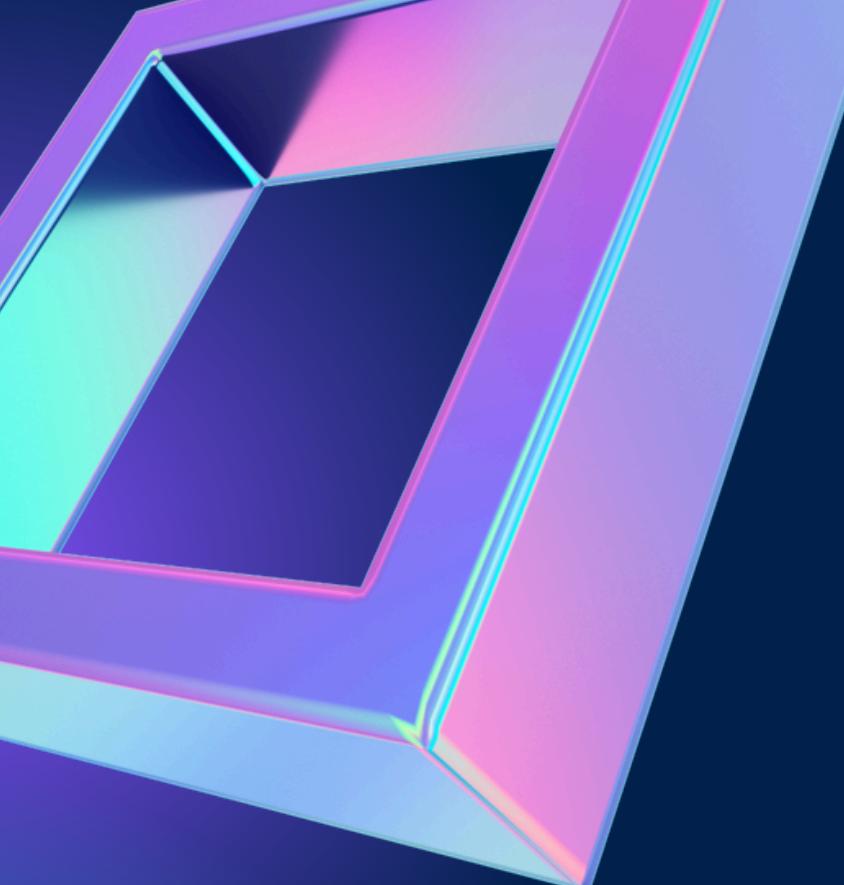
- Optimization Refinement & Testing

W9 – 10

- Testing
- GUI Development

W10 – 12

- Documentation and Presentation



Thank You for
listening