

Suite of algorithms & IoT-enabled tools, report

Project ref. no.	HORIZON-CL5-2024-D5-01-06 GA. N.º 101192375
Project title	Shifting to zero-emission logistics with right-sized, mission-focused, N1 eLCVs
Project duration	1st January 2025 – 30th June 2028 (42 months)
Related WP/Task	WP5 – Simulation & optimisation of eLCVs operations Task 5.3 – Development of software, algorithms, and IoT-enabled decision analytics tools
Dissemination level	PU - Public
Deliverable type	R - Document, report
Document due date	31/10/2026 (M22)
Actual delivery date	30/10/2026
Deliverable leader	CERTH (Centre for Research and Technology Hellas)
Document status	Final

Co-funded by the European Union

The Shift2Zero project has received funding from the European Union Horizon Europe Programme: project num. 101192375. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

Deliverable information sheet

Version	Date	Author	Document history/approvals
0.0	08/09/2026	CERTH	ToC and initial draft version
0.1	16/10/2026	CERTH	Draft version
0.2	23/10/2026	TOI, GRU	Cross reading process
0.3	29/10/2026	CERTH	Final Version
1.0	30/10/2026	Project Coordinator (EUT)	Submission

Executive summary

Project summary

Shift2Zero, Shifting to zero-emission logistics through right-sized, mission-focused, N1 e-LCVs.

Current market dynamics in EU reveal a gap between supply - existing N1 vehicles, and demand - evolving needs of urban logistics and climate targets. In 2023, 1.2M new LCV registrations were diesel-powered, and only 108,200 battery electric. Last-mile logistics, the least efficient and most complex part of the supply chain, presents significant opportunities for improvements at vehicle and operations levels. Dynamic requirements and increasing environmental impact require innovative solutions from the automotive industry, both from high volume OEMs and new entrants. S2Z aims to capitalize on the benefits of both vehicle platforms in the N1 segment - represented by IVECO's eDaily multipurpose platform, and Alke's ATX design-for-purpose platform - ultimately contributing to "Shifting to zero-emission logistics through right-sized, mission-focused, N1 e-LCVs".

To achieve this vision, S2Z proposes a 4-step user- and mission-centric design approach placing end-users and their needs at the core of all project activities. To this end, S2Z involves 5 LSPs & mobility operators as partners: Gruber, DHL, Diakinisis, Clem, DPD. As a result, S2Z will co-develop and shape at least 6 novel N1 concepts with enhanced and safe functionalities leading to tighter market fit, particularly in the segments of e-commerce, returns and cold deliveries.

Innovative concepts, from modular cargo bodies to vehicle control strategies with optimized tyres & brakes, as well as dual transport of people & freight, will be physically prototyped and tested in real-life operations in 6 pilot sites (Belgium, Greece, Italy, 2 in Norway, Poland).

S2Z brings a multidisciplinary consortium of 30 partners from 10 countries to cover the complete automotive and logistics value chains, complemented by policymakers to effectively ensure route to market: overcoming barriers for the adoption of S2Z eLCVs, reducing operational costs and environmental impact in scalable urban & sub-urban operations.

Executive summary of D5.3:

The transition to zero-emission urban logistics depends on how well new vehicles serve the demands of a delivery day. Each mission must accommodate the assigned cargo and meet customer commitments. It must also respect urban access conditions and remain feasible within the available battery energy and charging opportunities. These requirements connect vehicle design with operational planning. S2Z addresses this connection through right-sized, mission-focused N1 eLCVs and the digital tools needed to support their deployment. ***D5.3 presents the work carried out under T5.3 within WP5, translating the project's vehicle concepts and operational requirements into analytical capabilities for mission planning and performance assessment.***

The principal achievement is the development of a ***configurable analytical suite*** *that supports the preparation and assessment of eLCV operations. Its routing and loading components produce vehicle assignments and cargo arrangements that reflect delivery requirements. Energy models estimate battery demand throughout the mission, while charging algorithms determine how the required departure charge can be supplied. Decision analytics then convert the resulting operational quantities into*

performance indicators. Shared data structures and defined interfaces connect these components. Mathematical formulations and representative computational examples document their operation and provide a technical basis for subsequent integration and validation.

The work builds on the preceding WP5 activities. *T5.1 establishes the data requirements and models needed by the analytical components, together with the relevant KPI requirements. T5.2 defines the interoperability architecture and the arrangements for exchanging information between companies and digital services. T5.3 develops the algorithms and processing workflows that use this information. The resulting components support integration into the S2Z Urban Logistics Management Platform under T5.4 and scenario assessment under T5.5. They also connect the project's technical developments with pilot activities under WP6 and impact assessment under WP7.*

The vehicle innovations developed in S2Z raise new operational questions for LSPs (Logistics Service Providers). *A multi-temperature vehicle changes how deliveries can be combined within one mission. Swap box operations introduce decisions about replenishment and the timing of transfers. Energy-management technologies influence mission energy requirements, while geofencing determines where and when a vehicle may operate. Adopting these innovations therefore requires operational algorithms to be revised and extended to reflect their characteristics. D5.3 addresses this need by translating those characteristics into planning variables and constraints. This allows LSPs to assess how the new vehicle concepts fit their delivery requirements and how their operations may need to change to use them effectively.*

The architecture is designed to accommodate this variety without tying the analytical methods to a single pilot configuration. *Data acquisition is separated from analytical processing. Vehicle characteristics and operating conditions enter through configurable inputs, allowing the same models to be applied across different missions. Each engine has a defined responsibility and exchanges structured results through common interfaces. Components can operate independently or be combined according to the decision being addressed. Containerised services expose the analytical functions through APIs, while background processing and persistent job records support execution and traceability.*

Nine engines are organised into three functional blocks. *The Data and Operational Information Block prepares the inputs used by the Operations Planning Block. The third block connects decision analytics with simulation and scenario assessment. This structure follows the progression from understanding an operation to planning it and evaluating its performance. Within D5.3, the Simulation Engine is defined through its intended role and input/output relationships. Its detailed development and scenario applications belong to T5.5.*

The workflow begins with a consistent representation of the work to be performed. The ***Order Data Communication Engine*** *structures delivery demand and associates it with the relevant service and cargo requirements. The* ***Road Data Extraction Engine*** *adds road-based travel information and retains route geometry for subsequent assessment. Elevation is included where available. The* ***Vehicle Communication Engine*** *supplies common vehicle profiles and interfaces for incorporating telemetry.*

These services allow downstream models to use information from different sources while keeping source-specific formats separate from the analytical logic.

*The **Operations Planning Block** converts these inputs into missions that can be checked before dispatch. The **Routing Engine** assigns deliveries and determines the visit sequence. The **Cargo Loading Engine** checks whether the selected vehicle can carry and unload the assigned cargo. The **Vehicle Consumption Engine** estimates mission energy and assesses whether the battery reserve is maintained. The **Charging Scheduling Engine** determines how the required departure charge can be supplied within the available time and infrastructure. Their interaction is essential because the route must satisfy the corresponding loading, energy and charging requirements. The resulting assessments support mission preparation and comparison of alternative vehicle configurations.*

These dependencies also determine how a plan is revised. Changing a vehicle assignment can alter the available cargo space and the energy needed to complete the route. A different delivery sequence can affect unloading access and the time available for charging. The defined interaction logic allows a conflict identified by one engine to prompt a permitted mission revision, followed by reassessment of the affected results. Common identifiers keep each output associated with the correct mission revision. This preserves traceability as alternatives are examined. Completion and validation of the full coordinated workflow remain part of T5.4.

*The **Decision Analytics Engine** connects these calculations with operational performance. It processes planning outputs alongside available execution records to calculate indicators at mission, vehicle or fleet level. Delivery completion and punctuality can be examined against the resources used. Energy consumption can be related to distance and delivery activity, while charging results reveal readiness and infrastructure utilisation. Economic and environmental calculations use the applicable parameters and assessment boundaries. The analytical pipelines retain their assumptions and distinguish planned values from modelled estimates and observations. KPI functions are implemented at algorithm level, with platform reporting and pilot-specific configuration continuing under T5.4 in alignment with WP7.*

***The framework supports decisions at three horizons.** Daily Operations Planning prepares missions and provides the basis for comparison with observed execution. Fleet Requirements Analysis examines whether the available vehicles and charging resources can serve an anticipated workload. Strategic Network Planning allows LSPs and municipalities to assess changes to access policies or deployment arrangements together. Under T5.5, the planned simulation capabilities will extend these assessments to larger-scale scenarios. Baseline comparisons and sensitivity analysis will help identify the conditions under which particular configurations offer operational value.*

Representative computational examples and preliminary testing provide initial evidence of component functionality. They illustrate how the developed methods handle constraints and return results that can support subsequent decisions. Integrated computational performance and pilot impacts remain to be assessed. The immediate priorities are to complete orchestration and connect the relevant pilot data streams. Further work will configure user access and reporting, while operational

observations will support model calibration and validation. T5.5 will apply the analytical foundation to pilot configuration and larger-scale assessment, informing demonstration preparation under T6.1 and implementation under T6.2. The resulting evidence will also contribute to impact evaluation under WP7.

The current work establishes a ***stable architectural baseline for future extensions and for tools that can connect with the wider urban logistics ecosystem.*** *Updated traffic and curbside information could bring changing street conditions into mission reassessment. Further development could accommodate reverse logistics and more complex transhipment arrangements. Opportunity charging could extend planning beyond depot-based energy provision where suitable infrastructure is available. Connections to city digital twins could allow freight scenarios and operational results to be exchanged with broader urban models. These developments can build on the established interfaces while preserving consistency between the operational data and the decisions derived from them.*

Table of contents

Deliverable information sheet *2*
Executive summary *3*
Table of contents *7*
List of figures *9*
List of tables *10*
Terminology and Acronyms *11*
1. Introduction *12*
1.1 Objectives of the deliverable **12**
1.2 Relationship with Other WP5 Tasks **12**
1.3 Scope of the S2Z Analytical Suite **12**
1.4 Target Users and Operational Context **13**
1.5 Structure of the Deliverable **13**
2. Methodological Framework & State of the Art *14*
2.1 Scientific and Technological State of the Art **14**
2.2 S2Z Methodological Approach **17**
2.3 Project Innovations and Implications for the Analytical Suite **18**
2.4 Design Principles **21**
3. Solution Architecture *23*
3.1 Architecture Overview **23**
3.2 Data Sources and Main Data Flows **25**
3.3 Central Orchestration and Platform Integration **27**
3.4 End-to-End Operational Concept **29**
4. Engine Modules *33*
4.1 Overview of the Engines **33**
4.2 Data and Operational Information Block **33**
4.2.1 Order Data Communication Engine 33
4.2.2 Road Data Extraction Engine 35
4.2.3 Vehicle Communication Engine 36
4.3 Operations Planning Block **38**
4.3.1 Vehicle Routing Problem Engine 38
4.3.2 Cargo Loading Engine 45
4.3.3 Charging Scheduling Engine 50

4.3.4 Vehicle Consumption Engine....53
4.4 Analytics-Simulation and What-If Scenario Analysis Block....58
4.4.1 Decision Analytics Engine....58
4.4.2 Simulation Engine....60
4.5 Cross-Engine Dependencies and Interaction Logic....61
L1 Routing and Cargo Loading....63
L2 Routing and Energy Consumption....64
L3 Routing and Charging Scheduling....65
Shared constraints and mission revision....66
5. Conclusions and Extension Roadmap....69
5.1 Integrated Operational Features and Advanced Extensions....69
5.2 Summary of Deliverable's Achievements....71
5.3 Integration Readiness....73
5.4 Scalability and Transferability....73
5.5 Next Steps towards T5.4 and T5.5....74
References....75
Appendix A - Overview of the Reviewed Literature....79
Appendix B- Details of technical implications....81
Appendix C – Detailed Input and Output Specifications....82
Appendix D - Mathematical Formulations and Variable Glossary....91

List of figures

Figure 1: Methodological chain 18
Figure 2: Shift2Zero analytical architecture and data processing pipeline 23
Figure 3: Inter-engine interaction and feedback loops within the Operations Planning Block..... 24
Figure 4: Planning Horizons Workflow 29
Figure 5: Order Data Communication Engine Data Pipeline..... 34
Figure 6: Road Data Extraction Engine Routing & Elevation Workflow..... 35
Figure 7: Valhalla-generated road route with elevation profile and segment-level gradients in Thessaloniki..... 36
Figure 8:Vehicle Communication Engine Telematics Integration Architecture 37
Figure 9. Routing workflow from input data to feasibility assessment and module feedback. 39
Figure 10:Illustrative customer assignment and visit sequencing. Six customers each require one unit, and each vehicle has capacity for three units. 40
Figure 11 Illustrative routing adaptation under vehicle-specific geofencing. 41
Figure 12:Illustrative visit reordering to meet a geofencing time window. The departure, travel times and service durations are identical in both panels. 41
Figure 13. Illustrative consolidation of temperature-specific tours into one route using fixed compatible compartments. A denotes ambient goods, C chilled goods and F frozen goods..... 42
Figure 14. Illustrative replenishment at depot D in panel (a) and meeting point H in panel (b). Compatible equipment, two box positions and adequate payload are assumed.... 43
Figure 15. VRP and 3D loading operating scenarios. 44
Figure 16. Road routes from the saved run with 60 customers and four active vans. .. 44
Figure 17. Computed road-path adaptation around a configured restricted area. 45
Figure 18. Cargo loading workflow from input data to feasibility assessment and module feedback. 46
Figure 19 Single-temperature vehicles and a three-zone vehicle serving the same delivery demand. 48
Figure 20 Comparison of depot replenishment and Swap box transfer at an intermediate location. 49
Figure 21. Computed front-wall loading arrangement sample with 66 parcels. 50
Figure 22. Charging workflow from vehicle energy requirements to a checked schedule and planning feedback. 51
Figure 23. Charging and handling intervals recorded in the supplied output JSON example. 53
Figure 24: End-to-end computational pipeline 54
Figure 25: Mission discretisation 55
Figure 26:Outputs & Downstream Ecosystem 57
Figure 27:Simulation Engine Workflow..... 60
Figure 28: Operational Engine Dependencies and Orchestrator-Managed Revision Loops..... 62
Figure 29: L1 example. Cargo-loading conflict and revised vehicle assignments. Roads: © OpenStreetMap contributors. 63
Figure 30: L2 example. Energy-related routing revision and battery-feasibility assessment. Roads: © OpenStreetMap contributors..... 64

Figure 31: L3 example. Charging infeasibility and revised delivery assignment. Roads: © OpenStreetMap contributors. 66
Figure 32: Curbside-triggered route replanning with optional charging and operational data exchange. 71

List of tables

Table 1:S2Z methodological taxonomy linking analytical functions, engines, primary use, and decision outputs 18
Table 2: Adaptation of S2Z Operational Engines to Project Innovations 19
Table 3: Main S2Z Data Objects and Their Representations 25
Table 4: Implementation of S2Z Planning Directions Across the Three Planning Horizons 31
Table 5: Indicative Decision Analytics Outputs, Calculations and Visualisations 59
Table 6. Principal dependencies refreshed after an applied mission change 67
Table 7:Assessment of Task 5.3 Achievements Against Grant Agreement Commitments 72
Table 8: Summary of the reviewed literature and its relevance to the S2Z Analytical Suite 79
Table 9:Technical implications of S2Z innovations for the analytical engine modules 81
Table 10:VRP Engine input requirements 82
Table 11:VRP Engine outputs 82
Table 12:Cargo Loading Engine input requirements 83
Table 13:Cargo Loading Engine outputs 83
Table 14:Charging Scheduling Engine input requirements 84
Table 15:Charging Scheduling Engine outputs 85
Table 16:Vehicle Consumption Engine input requirements 86
Table 17:Vehicle Physical and Powertrain Input Parameters 87
Table 18:Vehicle Consumption Engine outputs 88
Table 19:Vehicle Consumption Engine Electrical Energy Output Components 89

Terminology and Acronyms

2D	Two-Dimensional
3D	Three-Dimensional
API	Application Programming Interface
BTM	Battery Thermal Management
CC/CV	Constant Current / Constant Voltage
CDS	Curbside Data Specification
CI/CD	Continuous Integration / Continuous Deployment
CO_2e	*Carbon Dioxide Equivalent*
CoG	Centre of Gravity
eHDV	Electric Heavy Duty Vehicle
eLCV	Electric Light Commercial Vehicle
EV	Electric Vehicle
EVRP	Electric Vehicle Routing Problem
GIS	Geographic Information System
GPS	Global Positioning System
gRPC	Google Remote Procedure Call
HiGHS	High-performance solver for linear optimization
HVAC	Heating, Ventilation, and Air Conditioning
IoT	Internet of Things
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LEZ	Low Emission Zone
LSP	Logistics Service Provider
MDS	Mobility Data Specification
MILP	Mixed-Integer Linear Programming
OSRM	Open-Source Routing Machine
PCM	Phase Change Material
PostGIS	Geographic Information System extension for PostgreSQL
REST	Representational State Transfer
S2Z	Shift2Zero
SoC	State of Charge
SoH	State of Health
UCC	Urban Consolidation Centre
VCE	Vehicle Consumption Engine
VRP	Vehicle Routing Problem
ZEZ	Zero Emission Zone

1. Introduction

This section introduces Deliverable D5.3 and defines its objectives, scope and role within WP5: Simulation & Optimisation of eLCVs Operations. It also explains the relationship between Task 5.3 and other WP5 activities, identifies the intended users and operational context, and outlines the deliverable's structure.

1.1 Objectives of the deliverable

This deliverable documents the development of the software, algorithms and IoT-enabled decision analytics tools under Task 5.3 of WP5, Simulation & Optimisation of eLCVs Operations. In line with the objectives defined in the Grant Agreement, the deliverable establishes a multipurpose and customisable analytical suite to support the efficient operation of eLCVs across different vehicle types, operational conditions and mission profiles. The suite includes tools for optimisation, simulation and operational decision support, covering functions such as vehicle routing, charging, cargo loading, energy consumption, vehicle communication, monitoring and forecasting. The deliverable describes the architecture, functionality, data interfaces, algorithmic approaches, implementation and technical validation of the developed components, providing the basis for their subsequent integration into the S2Z Urban Logistics Holistic Management Platform and their application in pilot and large-scale simulation activities.

1.2 Relationship with Other WP5 Tasks

Task 5.3 translates the data requirements, data structures and architectural specifications established through T5.1 and T5.2 into software, algorithms and IoT-enabled decision analytics tools. T5.1 identifies the operational, technical and economic data requirements, relevant data sources, data models and KPI requirements that define the information needed by the analytical components. T5.2 establishes the architectural and interoperability framework for data exchange, connectivity, APIs, IoT integration and interaction between digital solutions. Building on these inputs, T5.3 develops the analytical engines and software components for vehicle routing, charging, cargo loading, vehicle energy consumption, operational monitoring and forecasting. The task also defines the data inputs, outputs and interfaces required for communication between the individual engines. These components and interfaces provide the technical basis for T5.4, which integrates the tools and data streams into the S2Z Urban Logistics Holistic Management Platform. The resulting analytical capabilities and models also support T5.5, which applies them in pilot preparation and large-scale scenario simulations to assess different operational, policy and business conditions. This sequencing ensures that the analytical tools developed in T5.3 are firmly connected to the project requirements and pilot needs, while also ensuring their integration, operational use and subsequent validation within the wider WP5 framework.

1.3 Scope of the S2Z Analytical Suite

The scope includes the definition and implementation of the analytical engines, their functional architecture, input and output data structures, processing workflows and interfaces required for interaction between the individual components. The suite covers

tools for order and road data processing, vehicle communication, vehicle routing, cargo loading, charging scheduling, vehicle energy consumption, operational monitoring, forecasting and decision analytics. It also includes the integration of vehicle, operational, and contextual data required by these functions, together with the technical specifications and validation of the developed components.

The scope of D5.3 does not include the development of the S2Z Urban Logistics Holistic Management Platform itself, which is addressed in T5.4. Platform-level integration, user interface development and the final organisation of tools and data streams within the platform are therefore outside the scope of this deliverable. The deliverable also does not cover the development or execution of the large-scale pilot simulation activities assigned to T5.5. Pilot implementation, physical deployment of eLCVs and IoT equipment, collection of final pilot datasets and project-level impact assessment are also outside the scope of D5.3. These activities use the tools developed in T5.3 but are addressed through the corresponding WP5 tasks and the wider project work plan.

1.4 Target Users and Operational Context

The S2Z Analytical Suite is intended to support the operational and technical users involved in the planning, management and optimisation of eLCV logistics operations. LSPs and fleet or mobility operators use the analytical tools to plan missions, assess vehicle suitability, optimise routes and charging activities, and monitor operational performance. Pilot operators use the tools within the specific operational contexts of the Shift2Zero demonstration sites to support daily operations and evaluate alternative configurations. Planners use the analytical capabilities to examine vehicle, routing, charging, and cargo-related decisions under different operational conditions. Platform users access the available tools and analytical results through the S2Z Urban Logistics Holistic Management Platform as defined in T5.4. Technical and data users interact with the suite at the data and system level, including the configuration of inputs, management of data flows, monitoring of interfaces and interpretation of analytical outputs. The operational context includes urban and suburban logistics activities characterised by different vehicle types, mission profiles, cargo requirements, charging conditions, road environments and operational constraints across the ShiftZero pilot sites.

1.5 Structure of the Deliverable

Following this introduction, Chapter 2 reviews the scientific and methodological foundations of the analytical suite and their adaptation to S2Z innovations. Chapter 3 presents the solution architecture, data flows and operational concept, while Chapter 4 describes the individual engines and their interactions. Chapter 5 addresses advanced operational features, summarises the achievements of T5.3 and outlines the next steps towards platform integration and simulation activities. Supporting literature, technical implications of project innovations, detailed input and output specifications, and mathematical formulations are provided in the appendices.

2. Methodological Framework & State of the Art

This chapter establishes the methodological and scientific basis of the S2Z Analytical Suite by reviewing the relevant state of the art and defining the approach adopted within Task 5.3. It first examines the main scientific and technological domains addressed by the suite, then identifies the resulting requirements and gaps and defines how the analytical functions are combined to support operational planning, energy assessment, monitoring, simulation and decision analysis.

2.1 Scientific and Technological State of the Art

Achieving zero-emission urban logistics requires the coordinated application of optimisation, vehicle modelling, IoT-enabled monitoring and decision-support methods to address the operational challenges of electric light commercial vehicles (eLCVs). The development of the S2Z Analytical Suite draws on research in vehicle routing and optimisation, cargo loading, charging scheduling, vehicle energy consumption, telemetry, IoT-enabled fleet management and urban access constraints. This section reviews the main methodological contributions across these domains and identifies their relevance to the analytical requirements of S2Z. A detailed overview of the reviewed studies is provided in Appendix A - Overview of the Reviewed Literature.

Vehicle routing and optimisation. The electric vehicle routing problem (EVRP) extends conventional routing by incorporating battery-range limitations and recharging decisions (Kucukoglu et al., 2021). Goeke and Schneider (2015) model energy consumption as a function of speed, road gradient, and load, while Yang and Hyland (2024) examine fleet composition and battery-range feasibility. Froger et al. (2019) incorporate nonlinear charging functions, and Basso et al. (2019) integrate road-network energy estimation, including regenerative braking and vehicle mass. Shao et al. (2018) and Zhao et al. (2025) account for load-dependent consumption, with the latter also considering time-dependent speeds. Gao et al. (2024) address combined passenger and parcel transport, relevant to the S2Z dual-use concept. While these models match vehicles to missions primarily based on static volumetric capacity and battery range, they do not jointly account for modular body reconfiguration and payload-dependent energy demand. The S2Z VRP Engine addresses this gap by linking vehicle-platform right-sizing (e.g., IVECO eDaily versus Alkè ATX), modular body selection, payload-dependent energy consumption and multi-use mission profiles.

Cargo loading and stowage. Routing plans must also accommodate the physical and operational requirements of cargo. Pollaris et al. (2015) review routing with loading constraints, while Chi et al. (2026) focus on three-dimensional loading. Krebs et al. (2021) consider load bearing, stability, axle weights, and item accessibility, while Ferreira et al. (2024) link loading decisions to energy through weight-dependent emissions. Temperature-sensitive goods and reconfigurable cargo spaces introduce further complexity. Ostermeier et al. (2021) establish a multi-compartment routing framework, extended to rear loading of flexible compartments by Ostermeier et al. (2018). L. Chen et al. (2019) examine refrigeration energy during customer stops, while Masmoudi et al. (2026) jointly optimize vehicle selection, modular frigobox configurations, and routing. For electric cold-chain fleets, Wang et al. (2023) incorporate cooling requirements into routing, recharging and battery SoC calculations, while W. Chen et al. (2023) also

consider refrigeration demand and thermal cargo degradation. Vehicle reconfiguration is explored through swap-body routing (Huber & Geiger, 2017; Todosijević et al., 2017) and truck-and-trailer routing with access restrictions (Parragh & Cordeau, 2017). Together, these studies demonstrate how loading decisions extend beyond physical capacity to influence vehicle configuration, energy demand, and operational feasibility. Building on these contributions, the S2Z Cargo Loading Engine connects three-dimensional loading and modular multi-temperature configurations, including movable bulkheads and eutectic PCM plates, with evolving payload, battery consumption and route feasibility within a common planning workflow.

EV charging scheduling. Charging of urban freight fleets takes place mainly at the depot. Pelletier et al. (2018) schedule depot charging of electric freight vehicles under time-dependent prices, battery degradation and grid limits; Wu et al. (2023) minimise time-of-use charging cost subject to the number of charging points and the power limit of the station; Shi et al. (2023) include the CC/CV charging profile and degradation cost; and Mahyari and Freeman (2025) handle uncertain arrivals at a depot with heterogeneous multi-connector chargers through approximate dynamic programming. Pelletier et al. (2017) provide tractable battery degradation and charging models and note that most transport studies assume a fixed capacity depleted linearly with distance. These studies inform the S2Z Charging Scheduling Engine, although in them the energy each vehicle needs is typically an input rather than the result of its planned routes and payload. Depot charging models usually take the energy each vehicle needs as a fixed input (Pelletier et al., 2018; Shi et al., 2023; Wu et al., 2023). In S2Z, the Charging Scheduling Engine combines vehicle schedules with the energy requirements estimated by the Consumption Engine for the planned missions, subject to charger, depot and infrastructure constraints.

Vehicle energy consumption modelling. Electric freight operations require accurate mission-level energy estimates, accounting for distance, vehicle load, driving conditions, and auxiliary systems. Fiori and Marzano (2018) and Cieślik and Antczak (2023) examine payload effects using real-world driving data, while Łebkowski (2024) considers road topography and speed limits in predictive energy estimation. Di Martino et al. (2022) review thermal and auxiliary loads, while Miri et al. (2021) assess their effects on vehicle range through powertrain simulation. Environmental conditions further influence energy demand, as demonstrated by Rastani et al. (2019) for ambient temperature and HVAC use, and Osorio-Castañeda et al. (2025) for refrigeration during multi-stop deliveries. Although these studies address individual and combined factors affecting EV energy consumption, their integration into computationally efficient models supporting operational optimization remains limited. The S2Z Vehicle Consumption Model Engine addresses this gap through a unified, segment-level physics-based framework capturing vehicle dynamics, payload variations, road conditions and auxiliary loads, enabling battery feasibility assessment alongside routing, loading and charging decisions.

Vehicle telemetry and communication. Fiori and Marzano (2018) and Cieślik and Antczak (2023) build on measurements from electric delivery vehicles in real operation, and He et al. (2021) estimate battery state of health from real vehicle operating data rather than laboratory tests, so that usable capacity can be updated over the life of the vehicle. In S2Z, the Vehicle Communication Engine provides this link by turning telemetry into updated consumption parameters and battery capacity; data pipelines that feed

eLCV telemetry directly into routing and charging decisions are rarely addressed in the reviewed literature. To bridge this gap, S2Z aims to automate the calibration of energy models and battery health metrics from logged operational data, aiming to establish a feedback loop into dynamic routing and charging optimization.

IoT-enabled fleet management. IoT-enabled fleet management relies on real-time information to revise plans during operation. Mahyari and Freeman (2025) update depot charging decisions for charging service providers as vehicles arrive, Zhao et al. (2025) represent traffic conditions through time-dependent speed functions, and Huang et al. (2017) show that choosing road paths according to departure time and congestion saves cost and fuel and acts as a natural recourse in time-dependent congested networks. These elements correspond to the operational monitoring, KPI tracking, and performance evaluation capabilities of the S2Z Decision Analytics Engine and its integration with the Urban Logistics Holistic Management Platform; integrated IoT architectures for eLCV fleets are not covered by the reviewed works. Building upon these requirements, S2Z is designed to bridge real-time operational data with holistic platform monitoring, aiming to provide unified IoT-driven KPI tracking and adaptive decision support tailored for urban eLCV fleets.

Urban access and operational constraints. Municipal access regulations and spatial restrictions define where and when delivery vehicles can operate in urban centres. Muñuzuri et al. (2013) examine how delivery time windows affect routing and fleet requirements, while Bruglieri et al. (2025) incorporate Low Emission Zone (LEZ) regulations and charging availability. Beyond temporal and environmental restrictions, Ma et al. (2026) consider road weight and dimensional limits, while swap-body and truck-and-trailer routing studies (Parragh & Cordeau, 2017; Todosijević et al., 2017) further address customer accessibility constraints arising from vehicle configurations and maneuverability limits. Although these studies address individual access constraints and their operational implications, their integration with dynamic platform right-sizing and modular cargo configurations remains limited. The S2Z Geofencing and VRP Engine is designed to combine municipal regulations, weight restrictions, and delivery time windows, with vehicle platform sizing (IVECO eDaily vs. Alkè ATX) and modular body configurations, aiming to guarantee full spatial and regulatory compliance.

Taken together, the reviewed literature demonstrates substantial advances in routing, cargo loading, energy modelling, charging and operational monitoring, **but their integration into a common decision-support framework remains a key challenge**. The main contribution of the S2Z Analytical Suite lies in bridging these operational domains by connecting all digital engines within a single, unified workflow. By enabling continuous inter-engine communication and feeding logged operational data directly back into dynamic planning, S2Z establishes a fully integrated decision-support ecosystem tailored for right-sized, reconfigurable eLCVs across all project use cases.

Furthermore, **the existing approaches are not specifically designed around the technological innovations introduced by S2Z**. These include heterogeneous eLCV platforms, modular and reconfigurable cargo bodies, multi-temperature compartments with passive cooling technologies, integrated propulsion and thermal energy management, dual passenger–freight operations, and vehicle-specific urban access requirements. Accommodating these innovations requires more than combining existing

models, as their inputs, constraints and interactions must also reflect the characteristics of the S2Z vehicles and use cases. Task 5.3 therefore addresses both gaps by establishing a common analytical framework and adapting established modelling and optimization methods to the project's technological requirements, enabling coordinated assessment of vehicle selection, cargo configuration, routing, energy consumption and charging decisions.

2.2 S2Z Methodological Approach

The S2Z methodological approach combines established methods for logistics planning, vehicle energy modelling, charging, vehicle monitoring, and simulation and adapts them to the vehicle, cargo, energy, and urban-access requirements introduced by the project. These methods are implemented as interconnected engines within a common, modular workflow. This allows information and results to be exchanged between routing, loading, energy, charging, and analytical functions, so that changes in one decision dimension can be reflected in the others.

The methodology covers five complementary functions: data and context provision, planning and optimisation, energy feasibility assessment, execution monitoring and feedback, and simulation and decision analysis. Their use depends on the decision being addressed, from individual mission planning to fleet assessment and strategic scenario analysis. **The Order Data Communication Engine** structures operational demand, including orders, customer requirements, time windows, and parcel characteristics. **The Road Data Extraction Engine** provides the spatial and road-network context required for routing and energy assessment, including route geometry, travel information, and elevation where available. **The Vehicle Communication Engine** provides vehicle-specific technical and operational connector and, where available, telemetry and vehicle-state data. These functions primarily act as data enablers, transforming heterogeneous operational, GIS, vehicle, and IoT information into structured inputs for the analytical components.

The planning and optimisation functions use these inputs to construct operationally feasible missions. **The Routing Engine** determines vehicle assignment, stop sequencing, and route feasibility under spatial, temporal, capacity, and vehicle-related constraints**. The Cargo Loading Engine** assesses physical loading feasibility and determines suitable cargo configurations. **The Charging Scheduling Engine** determines charging activities by combining vehicle schedules and energy requirements with charger, depot, and infrastructure constraints. These functions primarily support Daily Operations Planning and Fleet Requirements Analysis, while their outputs can also contribute to higher-level scenario assessment.

The Vehicle Consumption Engine links operational planning to vehicle-specific energy behaviour. It estimates mechanical and electrical energy demand, auxiliary consumption, battery-state evolution, and mission feasibility using the planned route, vehicle configuration, payload, environmental conditions, and battery characteristics. Its outputs support energy-aware mission planning and comparison of alternative vehicle and operational configurations. During execution, vehicle telemetry and state information can be used to compare planned and observed performance, update vehicle-state estimates, and support autonomy assessment, calibration, and validation.

At higher planning horizons, **the Simulation Engine** evaluates alternative operating conditions, fleet configurations, and larger-scale scenarios, while **the Decision Analytics Engine** compares the resulting outputs using project-relevant performance indicators. These functions support Fleet Requirements Analysis and Strategic Network Planning by assessing alternative vehicle, fleet, infrastructure, access, and network configurations. The resulting methodological chain is illustrated in Figure 1.



Figure 1: Methodological chain

The same functions can be applied at different decision horizons. Daily Operations Planning uses current operational and vehicle information to support executable mission plans. Fleet Requirements Analysis uses historical or representative operational data to compare vehicle, cargo, and charging configurations. Strategic Network Planning uses simulation and analytical outputs to examine alternative hub, charging, access, and network configurations. This multi-horizon application reflects the intended multipurpose and customizable character of the S2Z analytical suite.

Table 1:S2Z methodological taxonomy linking analytical functions, engines, primary use, and decision outputs

Methodological Function	Engine	Primary Use	Main Output
Data & Context	Order Data Communication	Structure demand data	Demand input
	Road Data Extraction	Provide road and spatial context	Routing / energy data
	Vehicle Communication	Provide vehicle and telemetry data	Vehicle state
Planning & Optimization	VRP	Optimize vehicle assignment and routing	Route Sequence
	Cargo Loading	Optimize cargo allocation	Loading plan/Stowage
	Charging Scheduling	Optimize charging activities	Charging plan
	Vehicle Consumption	Assess energy use and battery feasibility	Energy / SoC / feasibility
Simulation & Analysis	Simulation	Test alternative and large-scale scenarios	What-if Analysis
	Decision Analytics	Compare scenarios and performance	KPIs / decision support / Intelligence insights

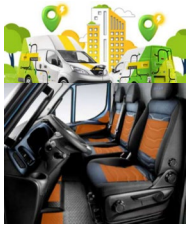
2.3 Project Innovations and Implications for the Analytical Suite

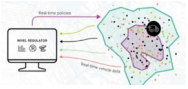
The methods reviewed in Section 2.1 provide the basis for the S2Z Analytical Suite. Their adaptation addresses the energy management, Swap box and modular cargo

configurations, multi-temperature cargo, and geofencing and access constraints. These adaptations concern both the algorithms and the information exchanged between engines. Table 2 summarises their implications for the operational engines.

Energy management links consumption to the vehicle, mission and operating conditions. The Vehicle Consumption Engine considers mass, road gradient, speed, acceleration and deceleration, drivetrain characteristics, regenerative braking, battery state and auxiliary loads. Changing payload therefore affects energy demand as well as capacity. Vehicle-specific battery, powertrain and charging parameters distinguish heterogeneous eLCVs. Low rolling-resistance tyres, braking and vehicle-control strategies are represented through their effects on resistance and energy recovery. Routing supplies route and payload information, while consumption assessment identifies energy constraints and charging requirements. Energy-feasibility results can exclude or penalise unsuitable routes, while the extended interface allows energy profiles and minimum-SoC locations to inform route assessment. Charging assesses whether the required departure energy can be supplied within the available opportunities. An unresolved charging requirement can therefore prompt reassessment of the mission..

Table 2: Adaptation of S2Z Operational Engines to Project Innovations

Project innovation	Routing Engine	Cargo Loading	Charging Scheduling	Consumption Engine
Multi-temperature cargo body 	Assign temperature-compatible vehicles. Retain the selected layout and cold-chain service requirements.	Place units in compatible compartments. Check physical fit and unloading access.	Use mission energy demand including applicable cooling auxiliaries. Account separately for depot pre-cooling and eutectic charging.	Use ambient temperature and the selected cargo configuration to estimate cooling and cabin auxiliary loads. Set compressor demand to zero for the S2Z eutectic configuration while retaining ventilation demand.
Swap box and transhipment 	Link box manifests to vehicles and visits. Synchronise donor and receiver transfer actions.	Check box dimensions and module fit. Track cargo custody and retained empty-box space.	Respect vehicle availability and transfer appointments. Refresh departure requirements after reassassignment.	Update payload by leg, including box tare. Account for waiting and handling energy.
Energy management and thermal comfort 	Assess battery-feasible missions. Revise assignments or timing when reserve or charging readiness fails.	Provide payload by leg. Supply load distribution where the configured energy model uses it.	Schedule compatible sessions within battery and depot power limits. Explicitly model any energy-export duties.	Use traction, regeneration and tyre parameters. Include cabin HVAC and other configured auxiliaries.

Project innovation	Routing Engine	Cargo Loading	Charging Scheduling	Consumption Engine
Geofencing and access control 	Apply vehicle eligibility and service periods. Refresh supported path, speed and distance restrictions.	Check the selected eligible vehicle's cargo space. Reassess loading after vehicle substitution.	Refresh availability and readiness when access rules change the mission. Retain compatible charging locations.	Recalculate changed paths, speeds and waiting. Assess detours and stop–go effects in the energy profile.

Swap box and modular cargo configurations require a more detailed representation than fixed vehicle volume and weight limits. The loading assessment must reflect the configured cargo space, module dimensions and compatibility with the assigned cargo and vehicle. The delivery sequence determines loading and unloading events and the resulting payload on each route leg. Loading checks whether the proposed arrangement is physically feasible, and consumption assessment uses the evolving payload to estimate mission energy and charging requirements. Alternative configurations can then be compared through simulation and decision analysis using capacity utilization, mission duration and energy demand.

Multi-temperature cargo adds compartment compatibility and thermal requirements to mission planning. Cargo must be assigned to suitable zones, with the selected layout, delivery sequence and mission duration reflected in loading and energy assessment. The generic consumption model accommodates conventional compressor-based refrigeration alongside eutectic thermal storage. For the S2Z pilot vehicle configuration using eutectic cooling, the compressor power term is set to zero, while ventilation and other applicable electrical auxiliaries remain included. Depot pre-cooling and charging of the eutectic plates are accounted for separately from battery demand during the route. Cabin HVAC remains a separate load. The resulting mission energy informs charging, while available temperature and operational observations support monitoring. Analytics and simulation can distinguish traction and thermal-system energy when comparing configurations.

Geofencing and access constraints make vehicle assignment and route feasibility dependent on geographical zones and vehicle-specific operating conditions. These include low-emission and zero-emission zones, loading areas, restricted roads and permitted access periods. Road information provides the spatial context, while vehicle characteristics determine eligibility and available position data support comparison of planned and executed operations. Restrictions can require a different vehicle, path or visit sequence, with corresponding effects on loading, travel time and energy demand. The same representation supports simulation and comparison of alternative urban-access policies.

These adaptations connect routing, loading, energy and charging through shared data and interfaces. A change in route sequence, vehicle assignment or cargo configuration can require the affected assessments to be repeated, while charging constraints can prompt mission revision. The same framework supports daily operations and assessment of alternative fleet, cargo, infrastructure and access configurations. Chapter

4 describes the engine methods and their interactions. Appendix B summarizes the technical implications of the project innovations.

2.4 Design Principles

The S2Z Analytical Suite applies five design principles that guide the implementation and integration of its analytical methods: modularity, interoperability, customisability, scalability and transferability. These principles determine how the individual engines, data structures, and interfaces can be configured and combined for different planning and assessment tasks.

Modularity.
The suite separates data acquisition and preparation from routing, cargo loading, energy assessment, charging scheduling, analytics, and simulation. Each module has defined responsibilities, inputs, and outputs, allowing independent development, validation, and updates within a common workflow. Modules can operate independently, sequentially, or through feedback loops. 3D-loading infeasibility can trigger VRP route resequencing, customer reassignment, or route splitting. Similarly, charging infeasibility can prompt reassessment by preceding planning modules rather than terminating the workflow.

Interoperability.
Data interfaces allow outputs from one engine to be used by others. The VRP Engine provides vehicle assignments, stop sequences, road routes, distances, and travel times. The Cargo Loading Engine adds loading configurations and payload evolution, which the Vehicle Consumption Engine uses to estimate energy demand and battery feasibility. Its energy and State-of-Charge results inform charging scheduling and other planning decisions. The Charging Scheduling Engine combines these results with vehicle and infrastructure data to determine feasible schedules. Structured machine-readable exchanges, including JSON-based interfaces where implemented, support communication between engines, external sources, and the S2Z platform.

Customizability.
The suite uses configurable parameters to accommodate different operational requirements and data availability. Vehicle and battery characteristics, cargo properties, working-time limits, routing constraints, charging infrastructure, and electricity tariffs are provided as inputs rather than embedded in the analytical logic. Routing can operate independently without detailed parcel dimensions, while sequential or fully coupled VRP–3D loading can be applied when sufficient information is available. Charging optimization can also prioritize electricity cost reduction or minimize time spent at high battery charge by completing charging closer to departure.

Scalability.
The modular structure allows computationally intensive procedures to be applied selectively. Engines can operate independently for simpler analyses or be coupled when detailed physical and operational validation is required. In routing and loading, faster procedures can be supplemented by detailed feasibility checks and fallback methods. Similarly, energy, charging, and simulation calculations can be restricted to relevant routes, vehicles, or scenarios. This supports larger demand sets and scenario assessments without solving the entire suite as a single optimization problem.

Transferability.
The analytical methods use common operational entities, including orders, customers, vehicles, parcels, routes, batteries, chargers, and infrastructure, rather than fixed pilot configurations. Local characteristics are introduced through datasets and parameters. The VRP Engine uses road-network, fleet and demand data; cargo loading uses parcel and compartment characteristics; vehicle consumption uses vehicle, battery, route and environmental information; and charging scheduling uses vehicle availability, charging requirements, infrastructure and electricity tariffs. The same analytical structure can therefore be adapted to different S2Z pilots, vehicle types and operating conditions without redesigning the core workflow.

3. Solution Architecture

Chapter 2 has established the methodological basis of the S2Z Analytical Suite and defined the main decision functions that need to be supported across different vehicle, mission and operational contexts. Building on this foundation, Chapter 3 describes how these functions are organised into a common technical architecture and how the different data, planning, energy and analytical components interact.

The solution architecture is designed as a modular, event-driven, and data-centric platform. It decouples core operational capabilities into specialized functional blocks, ensuring high scalability, seamless data flow, and resilient decision-making across fleet management pipelines.

3.1 Architecture Overview

Specifically, the S2Z analytical architecture depicted in Figure 2 is structured around three interconnected functional blocks: Data and Operational Information, Operations Planning, and Analytics, Simulation and Scenario Analysis. Together, these blocks establish a continuous data pipeline that transforms heterogeneous operational inputs into feasible mission plans, quantified energy and charging requirements, and KPI assessments. The architecture addresses operational environments in which data originate from multiple and heterogeneous sources, including fleet management systems, geospatial services, and IoT-enabled vehicle telematics. Data ingestion is separated from analytical processing, while individual engines remain modular and functionally distinct. This structure supports interoperability across pilot sites and software environments and allows analytical components to be reused without changes to their core decision logic.

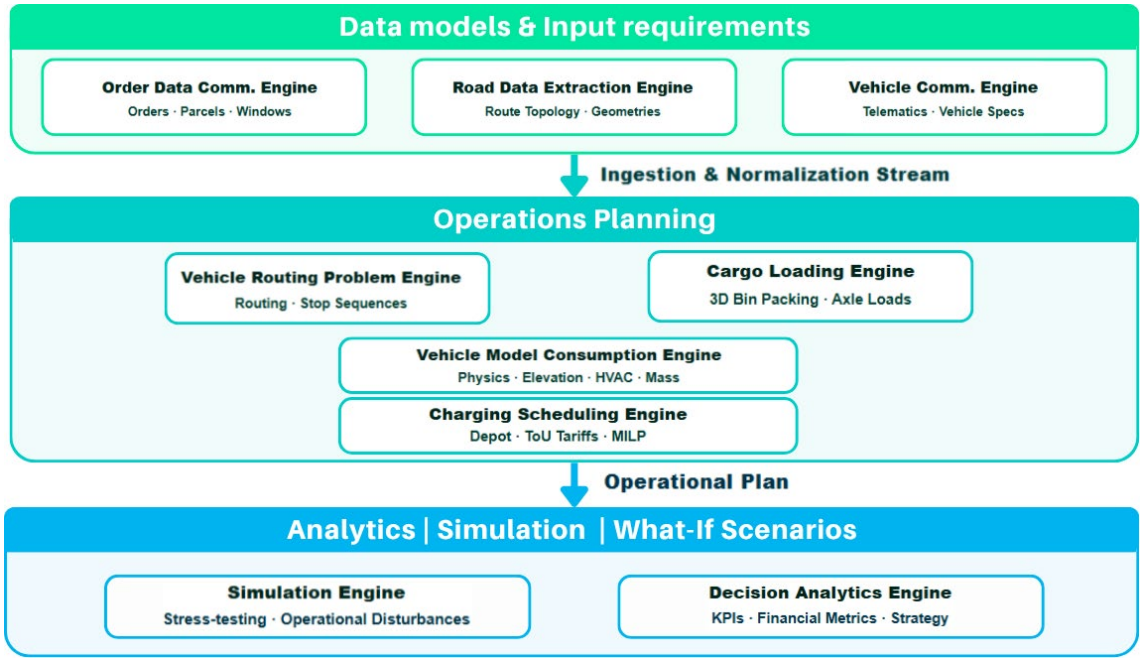


Figure 2: Shift2Zero analytical architecture and data processing pipeline

The foundation of this architecture is the Data and Operational Information Block, which provides the standardized data layer required by the downstream analytical engines. It integrates three core components. The Order Data Communication Engine ingests logistics demand, delivery time windows, cargo characteristics, and temperature requirements. The Road Data Extraction Engine provides spatial geometries, network topology, and elevation profiles used for routing and energy estimation. The Vehicle Communication Engine provides vehicle specification profiles and, where available, real time telematics data. These components operate as data transformation and communication services rather than optimization solvers. They therefore provide a clear interface between source systems and the analytical engines. This separation allows the same planning and assessment methods to be applied across different pilot implementations without modifying their underlying decision models.

Building directly upon this data layer, the Operations Planning Block converts structured operational context into optimized execution plans through four co-dependent decision engines as Figure 3 highlights.

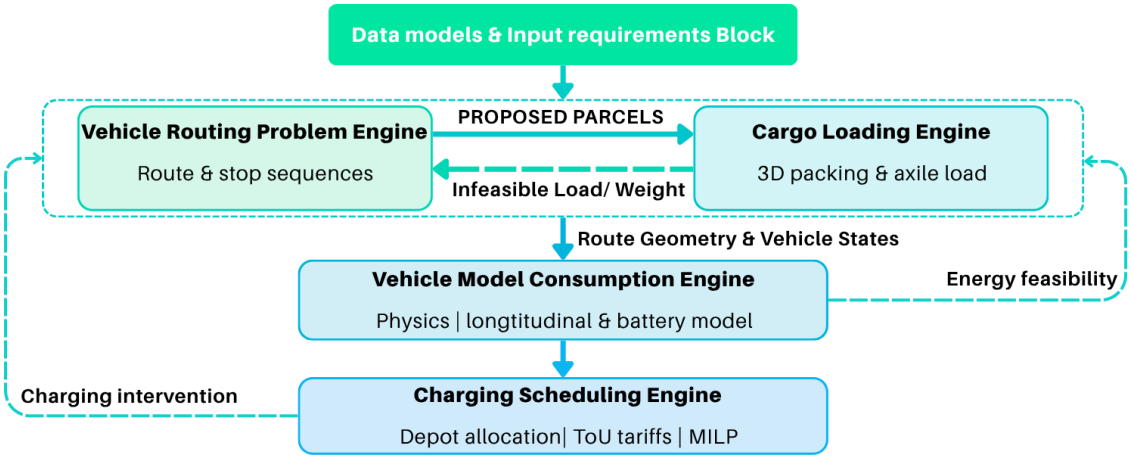


Figure 3: Inter-engine interaction and feedback loops within the Operations Planning Block.

The Vehicle Routing Problem Engine determines vehicle-to-order assignments and delivery sequences while considering spatial and temporal constraints. The Cargo Loading Engine evaluates the physical feasibility of each assignment through three-dimensional loading checks, axle weight distribution, cargo capacity limits, and temperature zone compatibility. When a loading or capacity constraint is violated, the resulting information is returned to the planning process to support reassignment or route adjustment.

The Vehicle Consumption Engine estimates energy use for each route segment using physics-based vehicle models. The assessment considers vehicle mass, road characteristics, speed, acceleration, elevation, and auxiliary energy demand. Auxiliary loads include cabin heating and cooling, cargo compartment temperature control, refrigeration, and other relevant onboard consumption. The resulting energy profile provides the basis for checking battery State of Charge margins along the planned mission.

The Charging Scheduling Engine uses the estimated energy requirements to determine charging schedules in relation to the available infrastructure. The scheduling process

considers charger availability, charging power, grid-related constraints, time-of-use tariffs where applicable, and vehicle departure requirements. The interaction between the four engines allows route feasibility, cargo feasibility, energy consumption, and charging requirements to be assessed as interdependent aspects of the same operational plan. Changes in one aspect can therefore be propagated to the related planning components.

To evaluate long-term performance and account for operational uncertainty, the Analytics, Simulation and What-If Scenario Analysis Block extends the analytical suite beyond deterministic daily planning. The Simulation Engine evaluates candidate plans using discrete event and stochastic modelling approaches. The simulations can represent operational conditions such as traffic delays, high or low ambient temperatures, charger unavailability, changes in demand, and battery degradation. This enables the assessment of plan robustness under conditions that cannot be represented adequately through a single deterministic planning run.

The Decision Analytics Engine combines execution data and simulation outputs to calculate relevant Key Performance Indicators. Depending on the pilot and assessment objective, these indicators can support analysis of operational performance, energy use, charging requirements, costs, and environmental impacts. The same analytical layer can support scenario analysis, Total Cost of Ownership assessment, and evaluation of alternative fleet electrification configurations across urban delivery networks.

Ultimately, these three blocks operate as a feedback-enabled framework rather than as a strictly linear sequence of processing steps. Operational data generated during execution, including vehicle telematics and charging information, can be used to update or calibrate the parameters applied by the consumption and planning engines. At the same time, results from simulation and decision analytics can inform subsequent planning cycles by identifying relevant operational constraints, sensitivity to disruptions, and charging infrastructure requirements. This bidirectional exchange between operational data, planning engines, simulation, and analytics provides the basis for the integration of the analytical suite into the wider S2Z digital environment and the Urban Logistics Holistic Management Platform.

3.2 Data Sources and Main Data Flows

The S2Z digital toolkit uses a common set of structured data objects to represent the main elements of the logistics operation. These objects cover demand, vehicles, cargo, depots and charging infrastructure, spatial conditions, route execution and analytical results. They provide the data interfaces between the Vehicle Routing Problem (VRP) Engine, Cargo Loading Engine, Vehicle Consumption Engine and Charging Scheduling Engine, while also supporting simulation and performance analysis.

The main domain objects are summarized in Table 3.

Table 3: Main S2Z Data Objects and Their Representations

Data Object	Purpose and Main Content
Depot / Hub	Location and operational characteristics of depots, UCCs and charging sites, including opening periods and charging availability/capacity.

Vehicle / Fleet	Vehicle identity, type and operational characteristics, including payload and cargo capacity, battery and powertrain parameters, and availability.
Order / Customer	Delivery or collection demand, location, service requirements, quantities, and time windows.
Parcel / Cargo Unit	Physical cargo characteristics, including dimensions, weight, handling requirements, temperature class, and delivery sequence.
Route / Stop	Vehicle tour and stop sequence, including planned arrival/departure times and operational activities.
Path / Road Segment	Road-level representation between consecutive stops, including geometry, distance, travel time, road characteristics, and elevation information where available.
Geofence / Spatial Constraint	Geographic zones and access rules used to represent LEZ/ZEZ areas, restricted roads, curbside conditions, and other spatial constraints.
Charging Request / Session	Charging requirements and availability, including vehicle SoC, required energy, charging window, and infrastructure constraints.
Telemetry / Execution Record	Time-dependent operational observations such as vehicle position, speed, SoC, energy use, charging status, and other available sensor or infrastructure information.
Analytical Output / KPI	Structured results generated by the engines, including routes, loading plans, energy and SoC estimates, charging schedules, operational KPIs, and scenario results.

The main engine interfaces use JSON based data structures to represent inputs and outputs. The JSON structures define the information each processing step requires and provide a common, machine-readable representation across the analytical components. Spatial information represents locations, routes, and geofences through point, line, and polygon geometries. Routing outputs provide distance and travel time information between relevant locations. Where available, road paths are enriched with elevation information for use in energy consumption modelling. Telemetry records provide time-dependent observations from connected vehicles and other available operational sources.

The data flow follows the dependencies between the analytical engines. Order, vehicle, depot, and spatial information define the operational planning context. The VRP Engine combines these inputs with routing information to generate vehicle assignments and stop sequences. The resulting route and path information is provided to the Cargo Loading Engine and Vehicle Consumption Engine. Cargo characteristics and vehicle capacities determine loading feasibility, while route characteristics, payload, and vehicle parameters support energy consumption and battery state estimation. The Charging Scheduling Engine uses the resulting energy requirements, along with vehicle schedules and charging infrastructure information, to determine feasible charging actions.

During vehicle operation, available telemetry and execution data can be associated with the relevant vehicle, route, and charging activity. This allows planned and observed

conditions to be compared and supports monitoring, model validation and calibration where applicable. The same structured representations can be used with historical or simulated data for fleet-level and strategic analysis.

The detailed data structures, input and output fields, and JSON representations are provided in the relevant technical appendices.

3.3 Central Orchestration and Platform Integration

The central orchestration layer connects Task 5.4 with the S2Z platform. The system is modular and containerized, with Vue 3, TypeScript for the frontend, ASP.NET Core for orchestration and Python services for the engines. The backend manages requests, jobs, engine communication, results and frontend notifications. This separation keeps logic independent from the platform interface, which allows the engines to progress while the backend maintains the integration contracts and the frontend receives common result structures.

To support this architecture, the frontend is built with the Vue/TypeScript toolchain and Vite and the ASP.NET Core backend is built through Docker. Python engine was used and is packaged with its dependencies and generated gRPC bindings. The service exposes its operations through gRPC using shared Protocol Buffer definitions, providing structured communication between Python and C#.

Likewise, optimization jobs, execution states, inputs and results remained through the backend database. Entity framework core migrations support schema evolution, while larger analytical artefacts such as route geometries or loading data can be stored in structured fields, related entities or referenced external storage.

Since the engine workflow depends on the complete service chain being available, readiness checks cover backend access to PostgreSQL/PostGIS and background processing, Keycloak authentication and communication with the Python service through gRPC.

In addition, execution is identifiable across the backend, worker, gRPC communication and Python engine through the optimization job identifier. Logs capture execution stage, operation, processing time and status, while metrics such as execution time, failures, retries and communication errors support monitoring of the engine workflow.

In practice, optimization requests enter through the authenticated REST API and are validated before a persistent job is created. Hangfire then executes the calculation asynchronously, the worker maps the request to the corresponding gRPC contract and the Python engine executes the selected function. Once finished, the engine returns a structured result or error to the backend, which validates and stores it. SignalR after this notifies the frontend, which retrieves the authoritative result through the REST API and updates the corresponding user facing view.

Throughout this workflow, the orchestration layer separates validation errors, infrastructure failures, model failures and infeasible scenarios. Temporary communication problems can be retried, whereas deterministic failures are stored as job failures. Timeouts, cancellation, restart recovery and duplicate request handling are also part of the job lifecycle.

Validation follows the same principle, as it is performed at both platform and engine boundaries. The REST API checks identifiers, formats and permissions, the orchestration layer maps the request to the correct engine contract and the Python service performs model-specific validation. Returned results are then validated as well, both before storage and before display.

Within this structure, the VRP workflow prepares vehicles, orders, depot, geofencing and travel information before execution. The VRP Engine produces vehicle assignments, ordered stops, route geometry, distances and travel times, while the Cargo Loading Engine evaluates vehicle capacity, cargo placement, utilization and unloading sequence.

Additional engines follow the same orchestration pattern, vehicle consumption can use route and vehicle information together with elevation, and climatic factors and charging scheduling can then use the resulting energy profile, vehicle availability and battery constraints. Each engine nonetheless remains independent and communicates through defined input/output contracts.

To keep these components consistent as they change, GitLab CI/CD exists as part of the project workflow and supports automated validation and integration of the components. The workflow includes frontend, backend and Python engine checks, gRPC contract verification, testing, application builds, and container image creation, providing a common validation path for platform and changes.

Complementing this, testing covers frontend utilities, backend services, engine functions, and shared gRPC contracts. Integration and end-to-end tests verify database persistence, engine communication, and background execution, and SignalR, the ASP.NET Core WebSockets-based package, publishes a status notification to the relevant authorized client or client group.

From a security perspective, sensitive configuration is kept outside source code and container images, while credentials and access to protected CI/CD resources are limited to the permissions required by each task. Dependency and container image checks provide additional validation of the software components.

To maintain stability, REST, JSON and Protocol Buffer interfaces are treated as shared contracts between the platform and engines. Changes should preserve compatibility where possible, while breaking changes should use explicitly versioned interfaces so different services can continue to communicate correctly.

For transparency, results should retain the engine version, scenario version, execution time, model parameters and relevant solver or library information. For routing, the routing-data source and version can also be recorded, supporting comparison and reproducibility of results.

Similarly, application, database and interface compatibility must be considered when changing versions. Validation of health, database connectivity, authentication, background execution, engine communication and result retrieval provides the basis for verifying a successful integrated version.

Overall, the current foundation includes containerized services, platform APIs, identity integration, background processing, persistent optimization job tracking, engine communication and frontend notifications. This already establishes the main path from

user request to execution and result display, while further automation, compatibility checks, observability and additional stages can build on the same architecture.

3.4 End-to-End Operational Concept

The S2Z platform is intended to support daily operations, fleet preparation and longer-term decisions about urban logistics. Figure 4 presents these uses through Daily Operations Planning, Fleet Requirements Analysis and Strategic Network Planning. Each addresses a different need. Companies organising deliveries need a plan they can put into practice and follow during execution. Fleet managers need to establish whether sufficient resources will be available for an expected workload. Municipalities and logistics companies considering changes to policies, vehicles or infrastructure need to assess their consequences before implementation. **The three concepts show how the same decision access point supports the end-to-end business intelligence pipeline progressing also to a systemic approach of designing urban logistics operations.**



Figure 4: Planning Horizons Workflow

In Daily Operations Planning, the process begins with the work that the company needs to perform and the conditions that apply to it. Municipalities provide access rules, delivery and loading windows, and area restrictions. Companies connect their order, fleet, cargo, depot, and charging data. The platform combines these inputs to support a workable operational plan and identify where assignments, resources, or operating arrangements need adjustment before dispatch. Operators review the plan to allocate work, prepare vehicles and cargo, organize charging, and communicate missions. The company then dispatches its vehicles. During execution, available vehicle data and operational records show progress against the plan. Delays, incomplete activities or changes in vehicle condition can prompt reassessment of the affected work, keeping planning and monitoring connected throughout the day. After missions are completed, statistics and performance indicators compare planned and observed results. Depending on data availability, these cover delivery completion, punctuality, vehicle utilization, energy use and charging activities. The findings help explain recurring delays,

capacity pressures, and performance differences, allowing operators to refine the assumptions and arrangements used in subsequent planning cycles.

Fleet Requirements (Asset Planning) Analysis uses the toolkit to prepare resources for an anticipated workload. An operator may, for example, need to determine how many vehicles, and which vehicle types, should be available for the following day. Known orders can provide the demand input directly. Where the workload remains uncertain, historical demand profiles or alternative assumptions can represent several possible situations. The assessment also considers city rules, service windows, cargo requirements, vehicle availability, and charging capacity, since these conditions influence how much work the fleet can perform.

The platform tools and planned simulation capabilities support comparison of candidate fleet arrangements under these conditions. The results help establish whether the available resources are sufficient, where bottlenecks arise, and how requirements change across the demand scenarios. Operators can examine estimated vehicle needs, expected workload distribution, charging requirements, and potential service shortfalls. They can then select an appropriate fleet arrangement, reserve additional capacity, or adjust the operating schedule before preparing the detailed daily plan. This mode also allows them to examine the consequences of a demand increase, reduced vehicle availability or a change in access conditions.

Strategic Network Planning extends the same approach to changes in the wider logistics system. Municipalities and companies can jointly examine a proposed low-emission zone, the adoption of multi-temperature vehicles or Swap box operations, and the introduction of transshipment points. These decisions affect several parts of the operation and may involve multiple companies, delivery areas and infrastructure providers. Co-design brings policy objectives and practical operating requirements into the definition of the scenarios, so that the alternatives assessed reflect the conditions under which they would be used. The process starts with a common baseline and a set of proposed changes. Municipalities contribute with partial rules, policy objectives and infrastructure options. Companies contribute with demand patterns, service requirements, and fleet characteristics. Each alternative describes how the change would work in practice. A multi-temperature scenario can examine the organization of deliveries with different temperature requirements. A Swap box scenario can examine transfers between vehicles and sites. An access-zone scenario can examine changes to vehicle eligibility, routes, and delivery times. Candidate locations and capacities for transshipment or charging facilities can also be compared. Results are compared with the baseline using relevant operational, economic, and environmental indicators aligned with the S2Z assessment methodology. Sensitivity analysis helps stakeholders understand how the conclusions depend on demand, fleet composition, infrastructure availability, or the level of innovation adoption. The outcome provides evidence to refine the scenarios and support decisions on policies, deployment arrangements, and investment priorities.

Table 4: Implementation of S2Z Planning Directions Across the Three Planning Horizons

Direction	Operational planning Daily operations	Tactical planning Asset requirements	Strategic planning Policy and deployment decisions
Main purpose	Prepare, carry out and review the daily delivery operation.	Estimate the number and types of vehicles and the charging capacity needed for an expected workload.	Co-design and assess changes to city rules, vehicle concepts and logistics infrastructure.
Municipality use	Provide access rules, delivery and loading windows, and spatial restrictions for daily planning.	Provide the rules and operating conditions used to assess the resources needed to serve demand.	Define policy objectives and candidate access zones or infrastructure options. Assess their consequences with logistics operators.
LSP use	Connect orders, fleet, cargo and depot data. Review the plan, dispatch vehicles, monitor missions and review performance.	Provide expected demand and asset availability. Compare fleet arrangements and prepare vehicles and charging resources for the next day or planning period.	Contribute demand patterns, service requirements and fleet options. Assess innovation adoption, collaboration arrangements and investment needs.
Platform use	Turn company data and municipal rules into workable mission plans. Link plans with available execution records and performance review.	Use planning tools and simulation to compare resource requirements under known or assumed demand and operating conditions.	Compare a baseline with alternative scenarios through planning tools, large-scale simulation and sensitivity analysis.
Main outputs	Mission plans, monitoring information, service and resource statistics, and records of deviations from the plan.	Estimated fleet mix, workload distribution, charging requirements, capacity bottlenecks and potential service shortfalls.	Scenario comparisons covering service performance, resource needs, energy demand and relevant economic and environmental indicators.

Decision and follow-up	Approve and dispatch the plan. Reassess affected work during execution and use observed performance in subsequent planning.	Select the assets needed, reserve additional capacity or adjust operating arrangements before finalizing daily plans.	Refine options and support policy, deployment and investment decisions. Examples include low-emission zones, multi-temperature vehicles, Swap boxes and transshipment points.

The matrix in Table 4 complements Figure 4 by showing how municipalities and LSPs use the platform for daily operations, asset planning and strategic decisions. For each scenario, it summarizes the users' roles, the platform's contribution, the main outputs and how these inform decisions and subsequent actions. Together, the figure and table explain how the three planning concepts are applied in practice.

4. Engine Modules

Building on the solution architecture defined in the previous chapter, this chapter describes the individual engines that implement the S2Z data, planning, energy, simulation and analytical functions. The engines are organised into three functional blocks, Data and Operational Information, Operations Planning, and Analytics, Simulation and Scenario Analysis, with defined interfaces supporting the exchange of information between them.

4.1 Overview of the Engines

The engines follow a modular architecture in which each component performs a defined function and exchanges structured information with other components. The Data and Operational Information engines primarily act as input providers and data transformation services. They do not perform optimization. Their role is to obtain information from external systems, convert it into the formats required by the S2Z analytical suite, and provide a consistent representation to downstream engines.

The Operations Planning engines implement the main decision logic of the analytical suite. The Vehicle Routing Problem Engine constructs and evaluates delivery plans. The Cargo Loading Engine checks whether the assigned cargo can be physically loaded into the selected vehicle. The Vehicle Consumption Model Engine estimates the energy required to execute the mission and evaluates battery feasibility. The Charging Scheduling Engine determines how and when the required energy can be supplied through the available charging infrastructure.

The Analytics, Simulation, and Scenario Analysis engines operate on planning results, operational data, and simulated missions. They support KPI calculation, performance assessment, and the evaluation of alternative operational configurations. This separation allows individual engines to be developed and extended without requiring changes to unrelated components.

4.2 Data and Operational Information Block

The Data and Operational Information Block serves as the ingestion, normalisation and contextualisation layer of the S2Z Analytical Suite. It connects pilot-specific order, geospatial and vehicle data sources with the common data representations used by the planning, energy and analytical engines. The block separates data acquisition and preparation from analytical processing, allowing source-specific formats to be transformed into consistent inputs for the downstream models.

4.2.1 Order Data Communication Engine

The Order Data Communication Engine provides the order information required by the S2Z planning workflow. Order data are obtained from the available pilot sources and combined with the relevant geofencing information. Since these data can be provided separately, the engine links the two representations so that each order is associated with its location and the applicable geographical information.

As illustrated in Figure 5, incoming scenario data are provided in structured JSON format and parsed into common representations of Order, Vehicle, Depot and Geofencing objects. The input is organized around deliveries, with each delivery associated with a customer identifier, an address or geographic coordinates, and one or more parcels. The same input also contains the main operational settings used to define the planning context, including working-day start and end times, available vehicles, service time per customer, and maximum route duration.

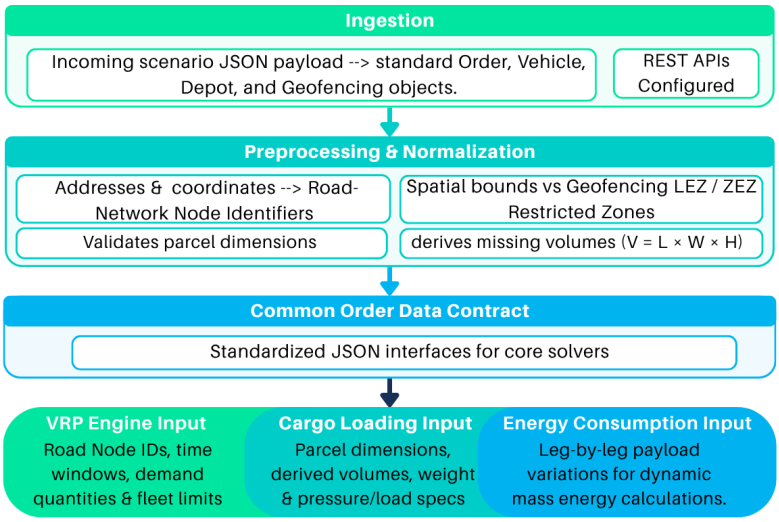


Figure 5: Order Data Communication Engine Data Pipeline

The order and parcel representations contain the information required by the downstream planning functions. Order attributes include geographical position, weight, dimensions, volume, temperature zone, customer identifier and service time window. Parcel attributes include parcel class, dimensions, weight and priority, together with relevant loading properties such as fragility, maximum allowable pressure and load-bearing capacity. Where volume is not explicitly provided, it can be derived from the available dimensions, providing a consistent volumetric representation for downstream cargo-loading calculations.

During preprocessing, the input data are validated and transformed into the internal representations used by the analytical workflow. Customer locations can be provided directly as coordinates or derived from the supplied address. In the current implementation, locations are associated with the drivable road network and enriched with the corresponding road-network node identifier. Each order is also evaluated against the available geofencing information and can be enriched with the identifiers of applicable restricted zones. Parcel and operational attributes are normalized before they are passed to the planning engines. The current VRP and 3D loading implementation also supports the mapping of parcel information to the predefined parcel classes used by the loading process.

The resulting order representation is used by the VRP Engine to define customer locations, demand, service windows, fleet availability, and relevant spatial constraints. The Cargo Loading Engine uses the associated parcel characteristics, including dimensions, weight, and loading properties, together with the vehicle cargo-space characteristics to assess physical loading feasibility. Relevant order and payload

information is subsequently available to the Vehicle Consumption Engine, where changes in payload affect vehicle mass and energy demand during the mission.

The Order Data Communication Engine therefore provides a common demand representation that is created once and consumed by the downstream engines according to their respective functions. The routing and loading results subsequently provide information for the energy and charging stages of the wider planning workflow.

REST APIs for demand data provision have been developed and configured and are available to the relevant data providers. In the architecture, pilot-specific order formats and geofencing sources are isolated within the communication layer and converted into the common S2Z representation without requiring changes to the core VRP, Cargo Loading or Vehicle Consumption models.

4.2.2 Road Data Extraction Engine

The Road Data Extraction Engine provides the road network and route information required by the routing and energy consumption models within the S2Z Analytical Suite. It interfaces with routing services, including OSRM and Valhalla, to retrieve road-based paths between relevant locations, such as depots and customer delivery points. The retrieved information includes route geometry, travel distances, and estimated travel times, allowing vehicle movements to be represented along the road network rather than through straight-line distances.

The engine processes the responses from these routing services and transforms them into a common S2Z route representation, ensuring that the downstream VRP Engine and Vehicle Consumption Engine can use the resulting information independently of the selected routing provider. For route planning, distance and travel time information supports the evaluation and optimization of vehicle routes. The corresponding geographical paths are retained for subsequent energy consumption assessment. Figure 6 illustrates this workflow, from retrieving road network information through OSRM or Valhalla to preparing route and elevation data for the downstream engines.

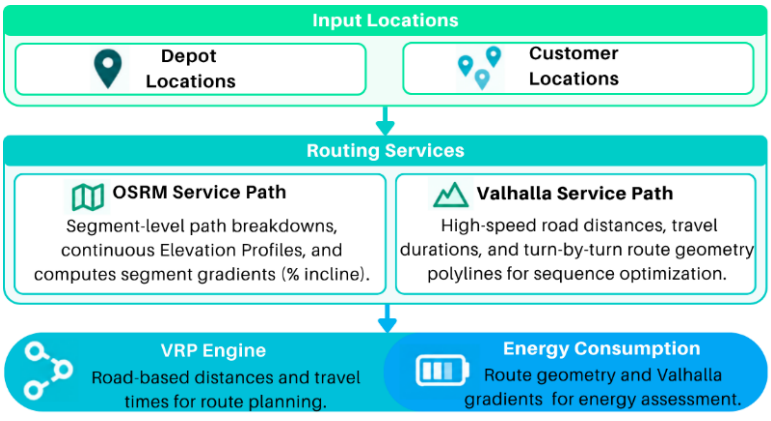


Figure 6: Road Data Extraction Engine Routing & Elevation Workflow

Where required, additional road attributes, such as elevation information available through Valhalla, can be incorporated into the route representation. These attributes support the reconstruction of the vehicle mission as a sequence of road segments,

including their distances and elevation changes. The Vehicle Consumption Engine uses this information, together with vehicle characteristics, payload conditions, and operating parameters, to estimate energy demand and battery State of Charge. Figure 7 illustrates an example of a Valhalla-generated route in Thessaloniki, showing the corresponding elevation profile and segment-level road gradients.

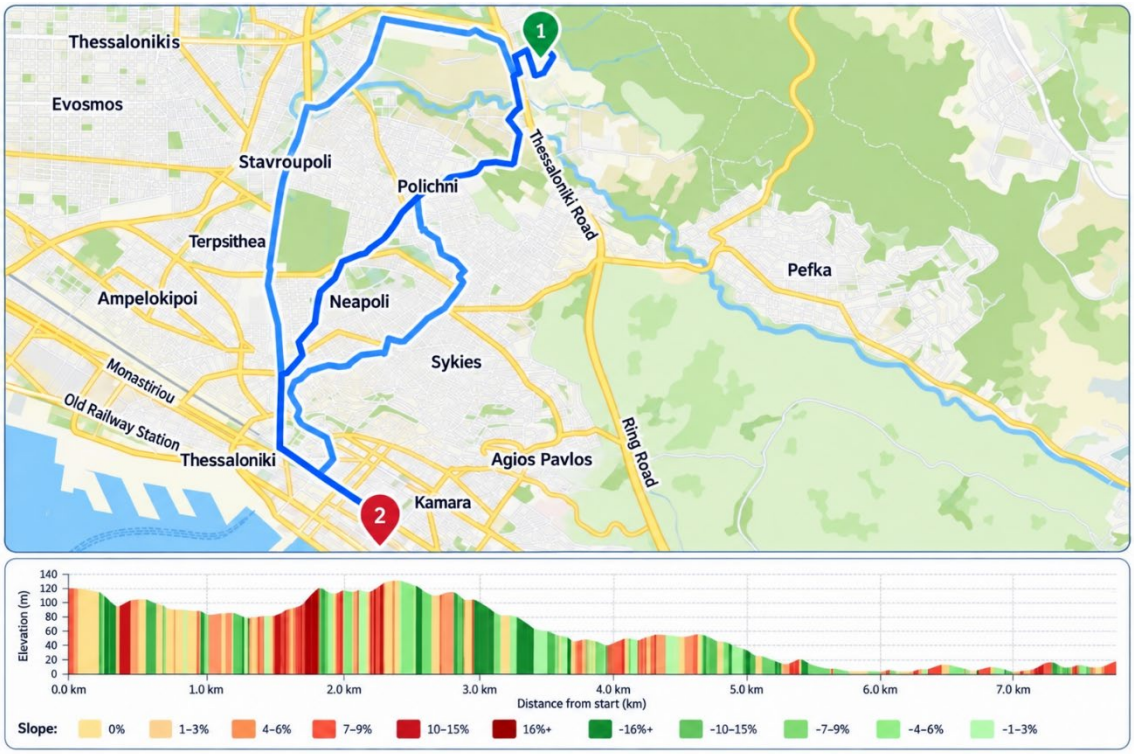


Figure 7: Valhalla-generated road route with elevation profile and segment-level gradients in Thessaloniki

The Road Data Extraction Engine therefore serves as a common source of spatial and routing information across the S2Z Analytical Suite. The VRP Engine uses road-based distances and travel times for route planning and assessment, while the Vehicle Consumption Engine uses route geometry and available elevation information for segment-level energy calculations. The resulting route information can also support simulation and subsequent analytical processing.

Current Implementation Status. The current implementation integrates OSRM and Valhalla as routing services and transforms their outputs into the common S2Z route representation. This provides the road distance, travel time, and geographical path information required by the VRP and Vehicle Consumption Engines, with additional attributes such as elevation incorporated where needed for energy assessment.

4.2.3 Vehicle Communication Engine

The Vehicle Communication Engine provides the link between vehicle-related data sources and the S2Z Analytical Suite. It supplies the vehicle information required to configure the analytical models and supports the incorporation of operational vehicle data during mission execution. The engine acts as a data provider and does not generate routing, cargo loading or charging decisions.

The architecture, presented in Figure 8, normalises vehicle information into a common vehicle profile containing the vehicle identifier, propulsion or fuel type, operational range, weight and volume capacities, cargo dimensions and cargo type. These characteristics are associated with the specific vehicle used in the planning workflow, transferred with the route solution and exposed to the operational dashboard. Vehicle information also contributes to contextual constraint evaluation. For example, propulsion characteristics can be used during geofencing preprocessing to determine vehicle eligibility for Low Emission Zones (LEZs) and Zero Emission Zones (ZEZs).

The target architecture extends the vehicle representation to real time operational information through the Alkè cloud platform. The API based integration provides available vehicle data such as vehicle status, State of Charge (SoC), speed, mileage, motor or controller parameters and GPS information. Depending on data availability, the vehicle state can also include State of Health (SoH), battery and vehicle temperatures and other operating measurements. These observations are associated with the corresponding vehicle profile so that dynamic vehicle state can be combined with the static characteristics used during planning. The available data can therefore be used to compare observed operating conditions with the assumptions applied during the initial mission assessment.

The planned communication interface maps the available Alkè data to the common S2Z vehicle representation while keeping the source specific telematics interface isolated within the communication layer. This allows the downstream analytical engines to use a common vehicle state representation independently of the source specific data structure. Where real time data are available, the vehicle state can be updated during mission execution. Historical vehicle data can also be retained for validation, calibration and subsequent analysis.

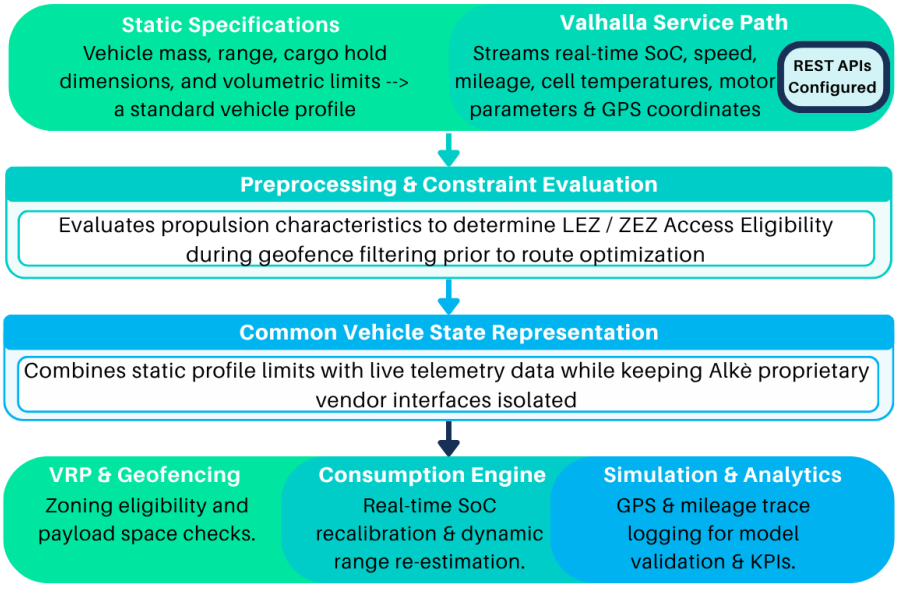


Figure 8:Vehicle Communication Engine Telematics Integration Architecture

The resulting vehicle information is used primarily by the Vehicle Consumption Engine, where vehicle characteristics and available state information are combined with route and payload data for energy and battery assessment. Vehicle state and position information can also support operational monitoring and, where implemented,

subsequent replanning when significant deviations from the planned mission are identified. The Simulation Engine and Decision Analytics Engine can use recorded vehicle data to compare planned and observed performance, support model validation and calibration, and provide inputs for scenario analysis. Vehicle position and propulsion information also support the contextual and geofencing constraints applied within the planning workflow.

REST APIs for vehicle data provision have been developed and configured to support integration with the Alkè cloud platform. The common vehicle profile and its use in contextual constraint evaluation form part of the current implementation, while the real time telemetry integration forms part of the target architecture. The availability and final use of individual telemetry variables depend on the data exposed by the connected vehicle platform.

4.3 Operations Planning Block

The Operations Planning Block is the decision-making layer of the S2Z Analytical Suite. It takes the order, vehicle, and road information prepared by the Data and Operational Information Block and applies dedicated optimisation and physics-based models to determine feasible routes, cargo arrangements, vehicle-energy requirements and charging schedules. The four engines in this block, Vehicle Routing Problem, Cargo Loading, Vehicle Consumption and Charging Scheduling, are kept as distinct components with defined interfaces between them, so each can be solved with the method suited to it while still exchanging feasibility information: a conflict identified in one engine, such as a loading or energy infeasibility, can be fed back to adjust the others. This supports an iterative cycle of plan definition, evaluation and refinement, and keeps the Operations Planning Block ready to supply outputs to the Analytics, Simulation and Scenario Analysis block that follows.

4.3.1 Vehicle Routing Problem Engine

The routing problem addresses the assignment of customer deliveries to the available vehicle fleet and the determination of the order in which each vehicle serves its assigned customers. Each route starts from the depot, visits a sequence of delivery locations and returns to the depot within the available operating period. The resulting routing plan must therefore satisfy both the delivery requirements and the operational limitations of the available vehicles.

To construct this routing problem, the model combines the validated customer and delivery information with the available fleet and the corresponding operational settings. Customer locations and parcel demand are provided through the structured order input, while the fleet definition specifies the available vehicles and their payload and cargo-volume capacities. The planning settings define the working period, service time at each customer and maximum route duration. These data are complemented by road-network information obtained through OSRM, which provides the pairwise road distances and travel times used during optimisation. A candidate solution is consequently represented as a set of ordered vehicle routes, with each route describing the sequence of customers served by one vehicle between its departure from and return to the depot.

Based on this representation, feasibility is determined by a set of operational constraints that restrict how customers can be grouped and sequenced within each route. Every

customer must be assigned to exactly one vehicle, while the total parcel weight and occupied cargo volume assigned to that vehicle must remain within its available capacities. The total duration of each route, including travel and customer service time, must also remain within the defined working horizon. Among the feasible alternatives, the optimisation gives priority to reducing the number of vehicles required to serve all customers and subsequently to reducing the total road distance travelled by the resulting fleet.

The processing chain in Figure 9 connects these inputs to route construction, feasibility assessment and, where enabled, loading feedback. Detailed input and output specifications are provided in Appendix C, Section C.1. The base mathematical formulation and variable definitions are given in Appendix D, equations (1)–(11).

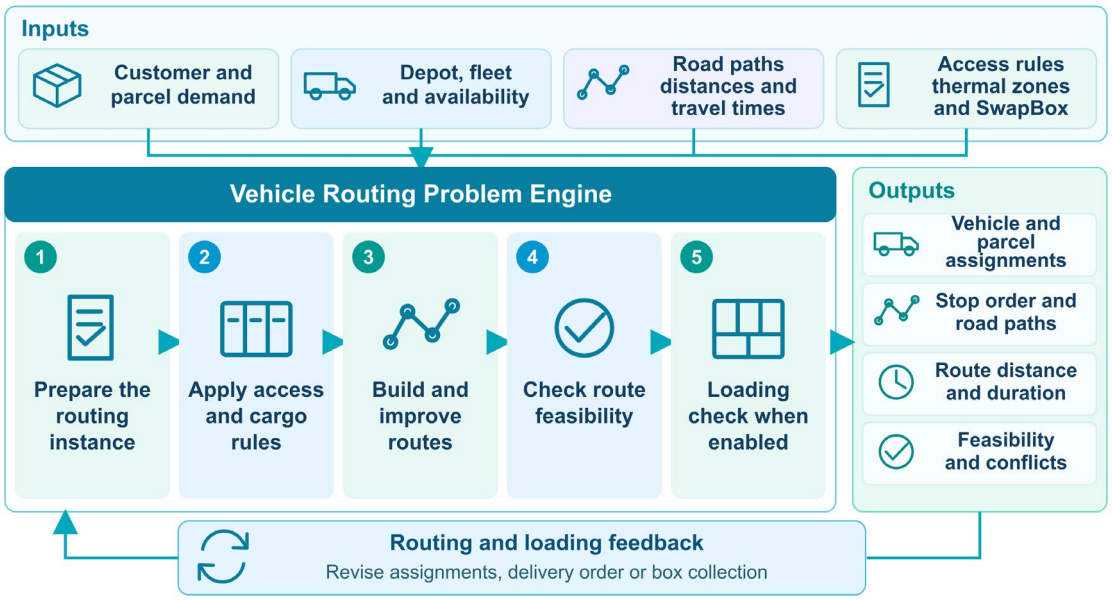


Figure 9. Routing workflow from input data to feasibility assessment and module feedback.

The route representation links three decisions. The model chooses which vehicles operate, which customers each vehicle serves and the sequence of visits. Equations (4)–(5) enforce mandatory assignment and vehicle activation. Depot connections and continuity follow (6)–(7), while (11) excludes disconnected customer cycles. Payload and volume are limited by (8)–(9), and travel plus service duration by (10). Volume is a routing-level check and does not establish three-dimensional fit. Figure 10 illustrates these decisions through two depot-to-depot routes. Each customer is assigned once, while demand remains within each vehicle's capacity.

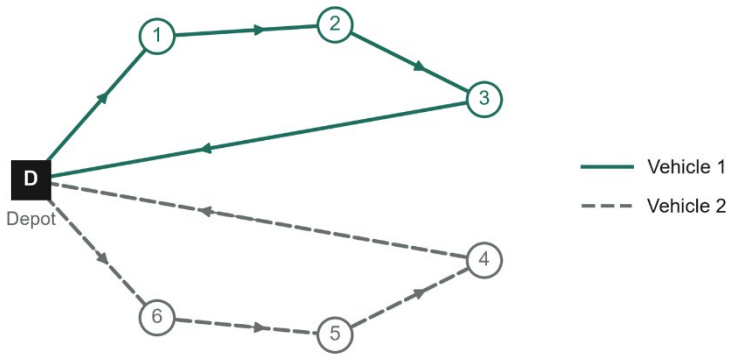


Figure 10:Illustrative customer assignment and visit sequencing. Six customers each require one unit, and each vehicle has capacity for three units.

Initial routes are improved through customer reassignment and changes to visit order. Adaptive large neighbourhood search removes and reinserts customer groups to test broader changes. Fleet consolidation checks whether the full demand can be served by fewer vans. Feasible alternatives follow the priority in (1)–(3), first reducing active vehicles and then road distance. Figure 16 shows routes from the saved run.

The base model supports three operational variants. Geofencing represents vehicle-specific and time-dependent geographical access restrictions. Multi-temperature transport uses fixed compartments for different temperature requirements. Swap box operations transfer complete loaded cargo modules between compatible vehicles. These variants change admissible assignments, paths or collection events while retaining the common customer-service requirements. The following relations show their main additions, with conditional implications expanded into linear constraints in the MILP.

Geofencing makes vehicle assignment and route choice depend on where and when a vehicle is allowed to operate. Restrictions may concern access by particular vehicle types, the use of specific roads or the permitted visiting periods at affected delivery locations. The routing plan may therefore require a different eligible vehicle, a legal detour or an adjusted visit sequence. Roads closed to every vehicle are excluded, while the actual distance and travel time of each selected alternative remain included in the optimisation. Vehicle dimensions and cargo configurations are also considered when determining access eligibility.

Figure 11 illustrates how vehicle-specific geofencing affects customer assignment and route planning. Ignoring access restrictions can produce a route that the assigned vehicle is not permitted to follow. In the illustrated example, restricted locations are served by an eligible vehicle, while the other vehicle follows a route outside the restricted area. This adaptation maintains service to all customers while respecting the area's access

conditions.

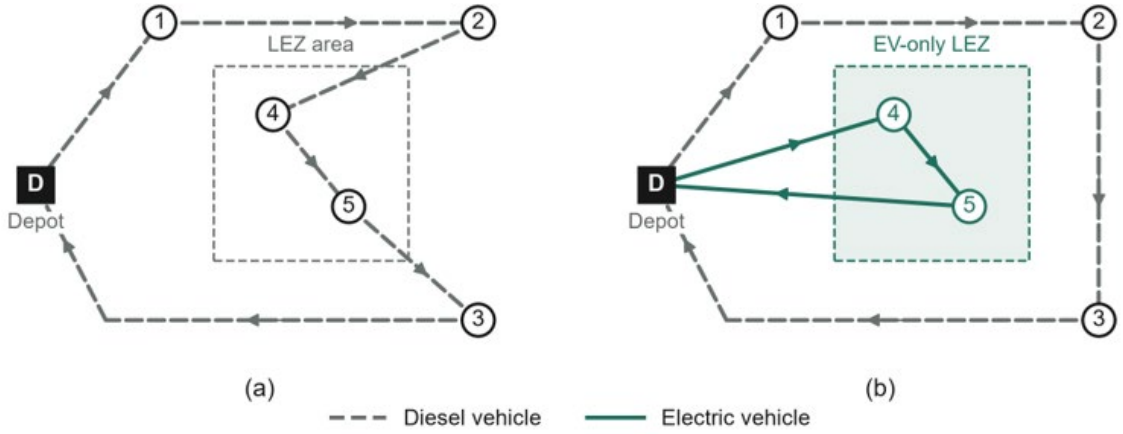


Figure 11 Illustrative routing adaptation under vehicle-specific geofencing.

Time-dependent geofencing can restrict when a specific delivery location may be visited. Figure 12 illustrates how time-dependent geofencing can affect the visit sequence. Ignoring the permitted access period results in a visit outside the allowed window. In the illustrated example, reordering the same customer visits brings the restricted stop within that window, preserving customer service while respecting the location's temporal access conditions.

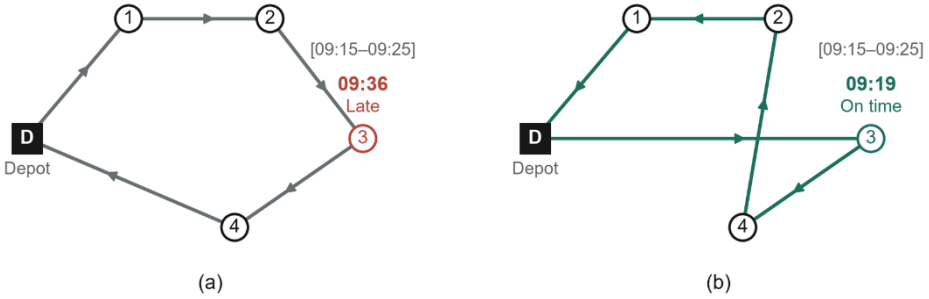


Figure 12:Illustrative visit reordering to meet a geofencing time window. The departure, travel times and service durations are identical in both panels.

For stops u, v and vehicle k, fixed set $\mathcal{P}_{uvk}$ contains legal road-path and departure-period alternatives indexed by h. Each alternative is valid for the full traversal and has an allowed departure interval $[a_{uvkh}, b_{uvkh}]$ in minutes from a common planning origin. Binary ξ_{uvkh} selects the alternative, base binary x_{uvk} selects the transition and continuous D_{uk} is departure time. The path-choice conditions are

$$\sum_{h \in \mathcal{P}_{uvk}} \xi_{uvkh} = x_{uvk}, \qquad \text{(G1)}$$

$$\xi_{uvkh} = 1 \Rightarrow a_{uvkh} \leq D_{uk} \leq b_{uvkh},$$

An empty alternative set prevents that transition. Arrival equals the preceding departure plus the selected travel time. Restricted visits must fit inside their permitted stay intervals, including any waiting. Windows governing service start alone are applied separately. Duration constraint (10) therefore includes elapsed travel, waiting, service and handling.

Objective (3) uses the selected legal-path distances, with fleet priority unchanged. Figure 17 illustrates the resulting detour around a configured area.

Multi-temperature transport allows goods with different temperature requirements to share an equipped vehicle. The three compartments have fixed boundaries and usable capacities. Routing assigns a parcel to suitable transport space, rather than treating every part of the cargo bay as interchangeable. Spare ambient space cannot replace an unavailable chilled compartment. As illustrated in Figure 13, separate tours for ambient, chilled and frozen goods can be consolidated into one shared route. This can reduce the number of vans required for mixed-temperature deliveries, provided each category fits its compatible compartment and the combined load satisfies payload and delivery-time limits.

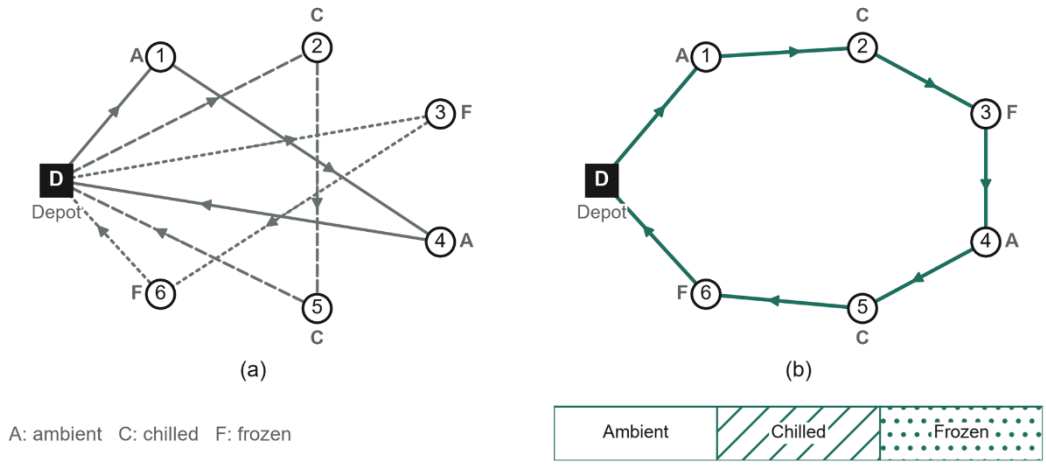


Figure 13. Illustrative consolidation of temperature-specific tours into one route using fixed compatible compartments. A denotes ambient goods, C chilled goods and F frozen goods.

Let I be the parcel set, c_i the customer of parcel i, Z_k the zones of vehicle k and v_i the parcel volume in m³. New binary g_{ikz} assigns the parcel to zone z. Fixed binary Γ_{ikz} records compatibility, and fixed V_{kz} is usable zone volume in m³. With base assignment $y_{c_i k}$ and activation u_k, routes loaded once at the depot satisfy

$$\sum_{z \in Z_k} g_{ikz} = y_{c_i k}, \qquad g_{ikz} \le \Gamma_{ikz}, \qquad \sum_{i \in I} v_i g_{ikz} \le V_{kz} u_k. \tag{T1}$$

These conditions add zone assignment and capacity to the base volume test (9), while retaining payload constraint (8). Actual fit is subsequently checked by loading. When combined with en-route collection, zone capacity applies to each simultaneous load rather than the all-day parcel total. The routing objective remains unchanged. Refrigeration demand is assessed by the consumption module.

Swap box operations allow a van to collect a prepacked removable cargo module at an eligible meeting point and then deliver its contents. The module is transferred as a whole from a larger vehicle, so collection can take place away from the depot without repacking its parcels during the handover. As shown in Figure 14, collecting the next box at a meeting point can replace an intermediate return to the depot for replenishment. The

same van continues its deliveries, subject to the donor appointment and sufficient payload and space for both the retained empty box and the new loaded box.

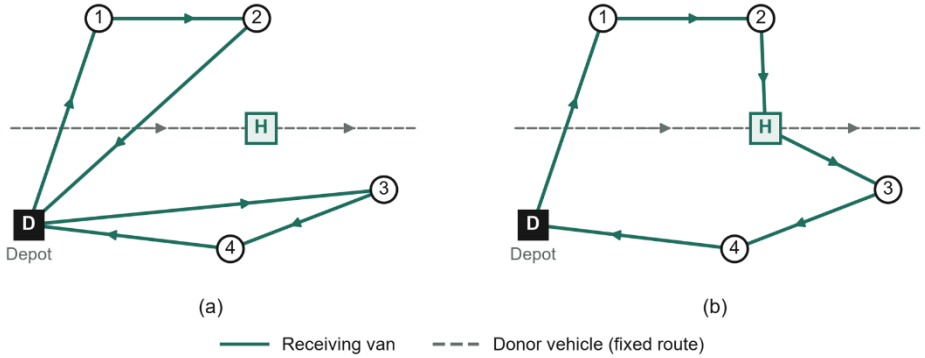


Figure 14. Illustrative replenishment at depot D in panel (a) and meeting point H in panel (b). Compatible equipment, two box positions and adequate payload are assumed.

Planning chooses the contents and receiving van for each box. Let B be the available boxes, K the available vans and I_{box} the parcels using boxes. New binaries $\alpha_{ib}, \gamma_{bk}, \beta_{bsk}$ and e_b denote parcel-to-box assignment, box-to-van assignment, collection at origin s and box activation. Fixed O_b contains permitted collection origins, including the depot only when allowed, and fixed binary κ_{bk} records box–vehicle compatibility.

$$\begin{aligned} \sum_{b\in B} \alpha_{ib} &= 1 \qquad (i \in I_{\text{box}}), \\ \sum_{s\in O_b} \beta_{bsk} &= \gamma_{bk}, \\ \sum_{k\in K} \gamma_{bk} &= e_b, \qquad \gamma_{bk} \le \kappa_{bk}. \end{aligned} \tag{S1}$$

Every used box has one receiving van and one collection origin. Additional links exclude empty active boxes and keep the contents with that van's customers. Transfer visits extend flow and subtour constraints (6)–(7) and (11). Parcel and box-presence variables require collection before delivery and retain the empty box until the depot. Internal capacity and temperature compatibility are checked separately.

At meeting s, fixed A_{sH}, D_{sH} are the larger vehicle's arrival and departure, with H labelling that vehicle. Continuous U_{sk} is transfer start, τ_{sk} the earliest time van k is ready, and D_{sk} its departure. Duration h_{sk} includes specified setup and per-box handling. Binary λ_{sk} indicates a transfer visit. Complete synchronisation requires

$$\begin{aligned} \lambda_{sk} = 1 \Rightarrow (\tau_{sk} \le U_{sk}, \quad A_{sH} \le U_{sk}), \\ \lambda_{sk} = 1 \Rightarrow (U_{sk} + h_{sk} \le D_{sk}, \quad U_{sk} + h_{sk} \le D_{sH}). \end{aligned} \tag{S2}$$

Transfers fit wholly inside both vehicles' presence periods and cannot overlap when sharing one handling resource. Capacity limits (8) – (9) apply after every collection, delivery and intermediate transfer, including on the larger vehicle. Mass includes box tare, remaining contents and background cargo. Volume counts external boxes once, without adding internal parcel volumes. Empty Swap boxes retain their tare mass in the payload calculation. They are assumed to be folded after emptying and carried during the remaining route without creating a binding cargo-space constraint.

Meeting travel enters (3) and waiting and handling extend (10). Fixed larger-vehicle mileage contributes to system kilometres without changing van-plan rankings.

The output contains the active fleet, vehicle and parcel assignments, ordered stops, complete road geometry, route distance and duration, and feasibility checks. Routing can operate independently without 3D certification, pass a completed plan to loading, or exchange results with loading repeatedly. A loading conflict can trigger a revised assignment, delivery order or box collection while preserving mandatory service. The resulting route and payload information also supports consumption assessment. The three-dimensional loading interface is detailed in Appendix C, Section C.2. These three modes of operation are summarised in Figure 15.

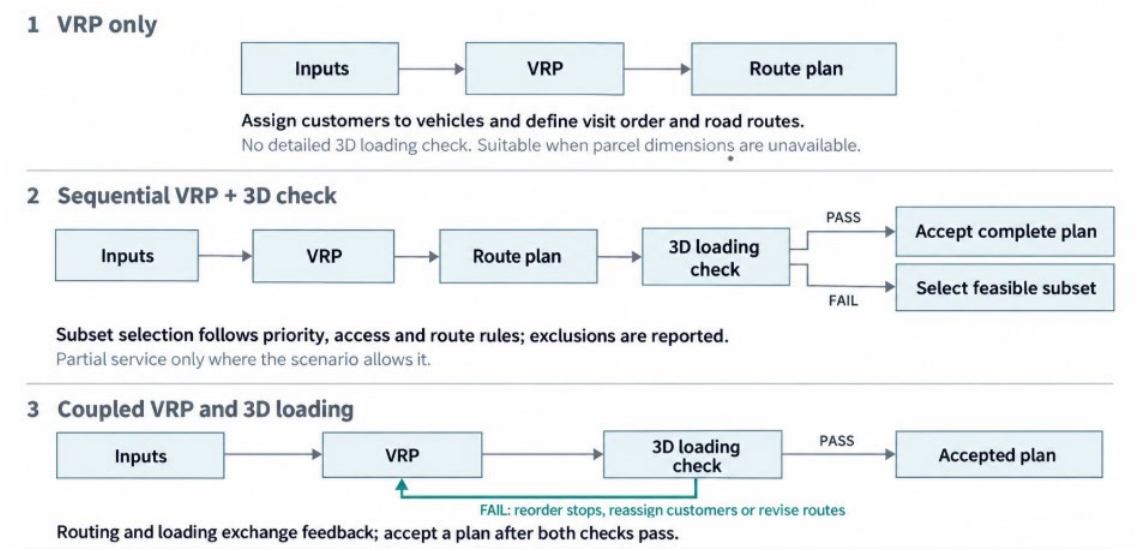


Figure 15. VRP and 3D loading operating scenarios.

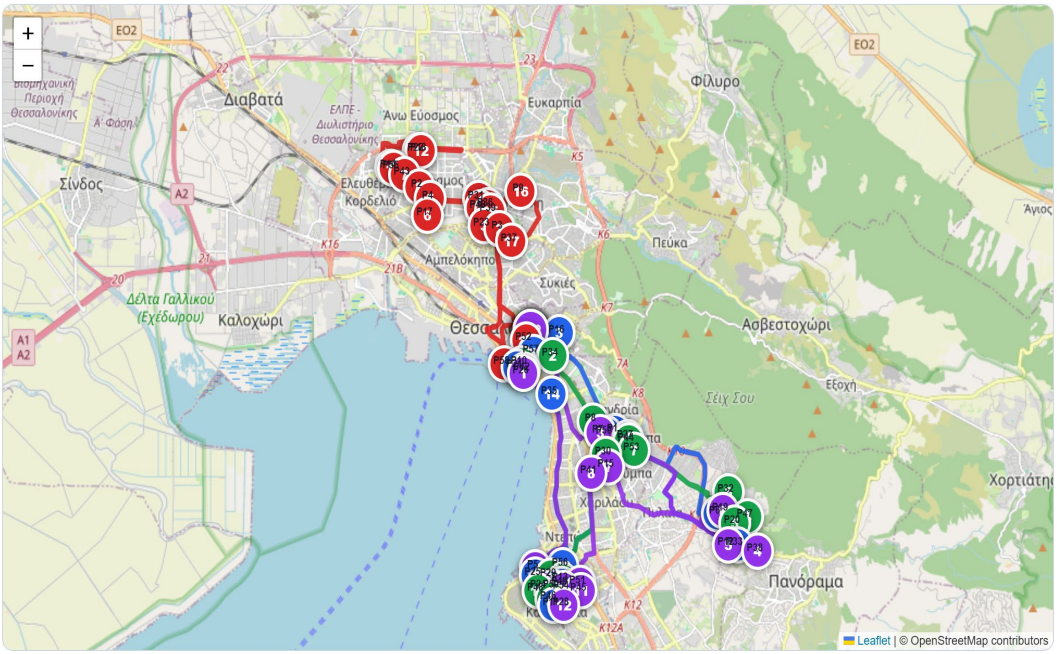


Figure 16. Road routes from the saved run with 60 customers and four active vans.

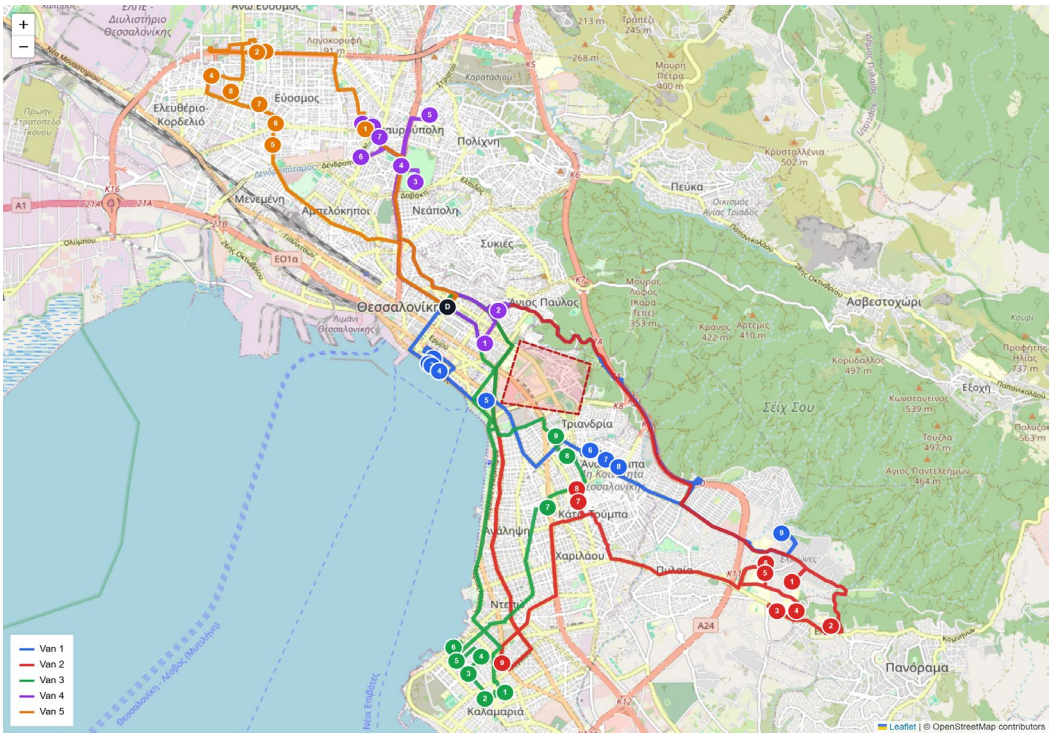


Figure 17. Computed road-path adaptation around a configured restricted area.

4.3.2 Cargo Loading Engine

The Cargo Loading Engine complements the routing process by evaluating whether the delivery plan generated by the VRP can be realised within the physical cargo space of each assigned vehicle. While the routing stage determines which customers are served by each vehicle and in which order, it does not by itself guarantee that the corresponding parcels can be arranged safely and practically inside the cargo compartment. The loading stage therefore introduces the spatial, structural, and unloading-related constraints required to convert a route-level assignment into a physically executable vehicle mission.

To perform this assessment, the loading model combines the routing output with detailed parcel and vehicle information. For each vehicle, the VRP provides the assigned customers, associated parcels, and planned stop sequence, which establishes the order in which the cargo must become accessible during delivery. Each parcel is described by its dimensions, weight, and stackability information, while the vehicle representation defines the cargo-bay dimensions, maximum payload, empty-vehicle axle loads, and the parameters required to evaluate weight distribution. These inputs define the three-dimensional loading instance associated with each proposed route.

Given these inputs, each parcel must be assigned an admissible three-dimensional placement and orientation within the cargo bay. A feasible arrangement must remain within the vehicle boundaries, prevent spatial overlap, provide adequate support for elevated parcels, and respect the categorical stackability rules. The delivery sequence introduces an additional operational requirement because parcels belonging to earlier delivery batches must remain accessible without excessive rehandling. The resulting

configuration must also satisfy the vehicle payload and axle-load limits, both at departure and after successive customer unloadings.

Figure 18 shows how the loading instance is constructed, searched, and checked before results are returned to routing. The data exchanged through this interface are specified in Appendix C, Section C.2. The base placement model, variable definitions, and loading objectives are presented in Appendix D, equations (12)–(24).

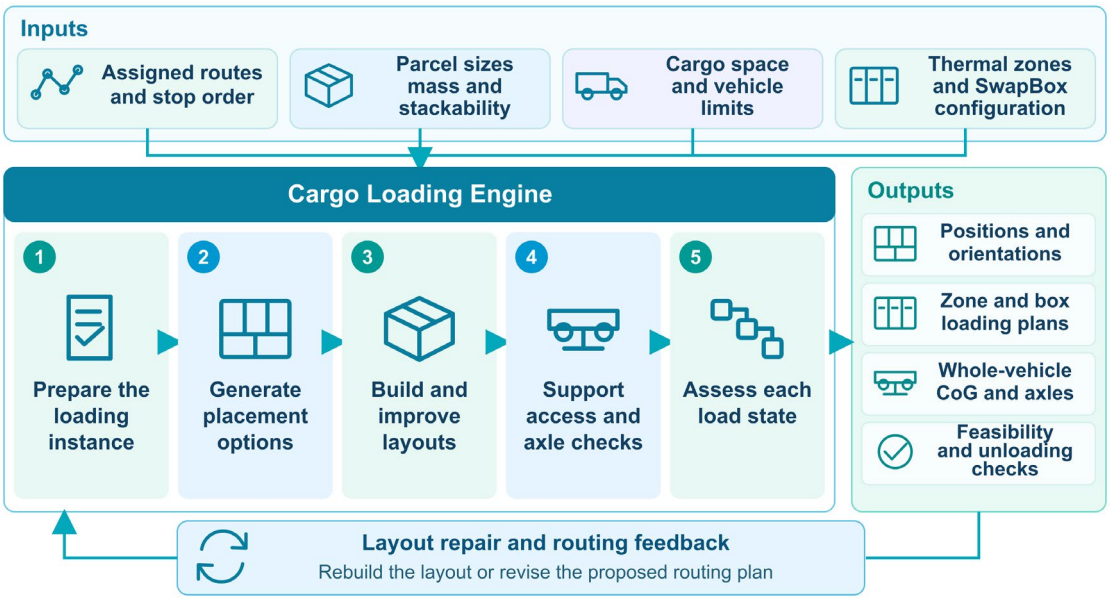


Figure 18. Cargo loading workflow from input data to feasibility assessment and module feedback.

The model represents each parcel through admissible placement candidates. A candidate combines a position and an allowed orientation. Exactly one is selected for each assigned parcel under (15). The boundary conditions in (14) and incompatibility conditions in (16) keep parcels inside the available space and prevent overlap. This tests physical fit beyond the aggregate volume available to routing.

An elevated parcel must receive at least 75% support of its base area under (18). The stackability condition (17) prevents non-stackable parcels from supporting another parcel. Centre-support and overhang conditions appear in (19). Support chronology (20) prevents the lower parcel from being delivered before the parcel resting on it. These checks preserve support as deliveries progress without requiring parcel-pressure data.

Unloading follows the customer sequence. Direct and indirect blockers activate movement indicators under (21). Let I contain all parcels, I_c the parcels delivered to customer c, and fixed $\overline{y}_{ck}$ indicate assignment to van k. Binary $h_{jck,\mathrm{move}}$ indicates that another parcel j must be displaced. Fixed H is the allowed number of distinct displaced parcels for one complete customer unloading. The base bound (22) is retained as

$$\sum_{j \in I \setminus I_c} h_{jck,\mathrm{move}} \leq H \overline{y}_{ck}. \tag{L2}$$

A parcel counts once across the whole van during that customer's unloading, even if several blockers identify it. The allowance does not restart for every target parcel or box. Departure and remaining-load axle limits follow (23)–(24). Centre-of-gravity assessment

uses the complete empty-vehicle reference and the actual cargo positions, rather than the cargo-bay centre.

Construction follows reverse delivery order, placing later deliveries deeper where feasible. Extreme-point placement tests positions and rotations at boundaries formed by existing parcels. Alternative constructions and destroy-and-repair search rebuild difficult parts of the layout. Selective finite-candidate optimisation tests combinations of predefined placements. Among feasible alternatives, objective (12)–(13) prioritises unloading effort, complete-vehicle centre-of-gravity deviation and occupied-space compactness.

The operational variants change the geometry or load states checked by this process. Fixed thermal compartments reserve suitable space for each cargo category. Swap box requires a linked arrangement inside the removable module and on its carrying vehicles. Geofencing changes loading indirectly when it requires another vehicle or delivery order. The corresponding base constraints are retained and adapted to the resulting instance.

A geofencing-driven reassignment may replace the cargo bay or change which parcels must be accessible first. Loading consequently rechecks bounds, support chronology, unloading access, and remaining axle loads under (14)–(24). Geofencing adds no separate parcel-separation rule and does not change the loading objective. Restrictions on actual vehicle mass use the relevant load state.

Multi-temperature loading turns the zone assignments of Section 4.3.1 into a physical arrangement. The complete parcel must lie inside its compatible compartment, with partitions, doors, and clearances included. The result identifies where each cargo category fits and whether it can be reached in delivery order.

For zone z of vehicle k, fixed lower bounds X_{kz}, Y_{kz}, Z_{kz} and upper bounds $\overline{X}_{kz}, \overline{Y}_{kz}, \overline{Z}_{kz}$ define the usable space in metres. Candidate a for parcel i has fixed lower coordinates $x_{ika}, y_{ika}, z_{ika}$ and oriented dimensions $l_{ika}, w_{ika}, h_{ika}$ in metres. Set A_{ikz} contains compatible candidates satisfying these bounds. Binary p_{ika} selects a placement and g_{ikz} is its zone assignment from (T1).

$$\begin{aligned} X_{kz} \le x_{ika}, \quad & x_{ika} + l_{ika} \le \overline{X}_{kz}, \\ Y_{kz} \le y_{ika}, \quad & y_{ika} + w_{ika} \le \overline{Y}_{kz}, \\ Z_{kz} \le z_{ika}, \quad & z_{ika} + h_{ika} \le \overline{Z}_{kz}, \\ \sum_{a \in A_{ikz}} & p_{ika} = g_{ikz}. \end{aligned} \tag{T2}$$

These relations strengthen the vehicle-wide boundaries in (14) and replace the placement condition (15) with a zone-specific choice. Support and blocking relations in (17)–(22) are rebuilt for the actual partitions and openings. A partition is not assumed load-bearing without equipment data, and support cannot pass through it. Zone boundaries are fixed, and objective priorities remain unchanged.

The multi-temperature extension combines temperature compatibility with compartment capacity and unloading access, helping assess whether different cargo categories can share one vehicle.

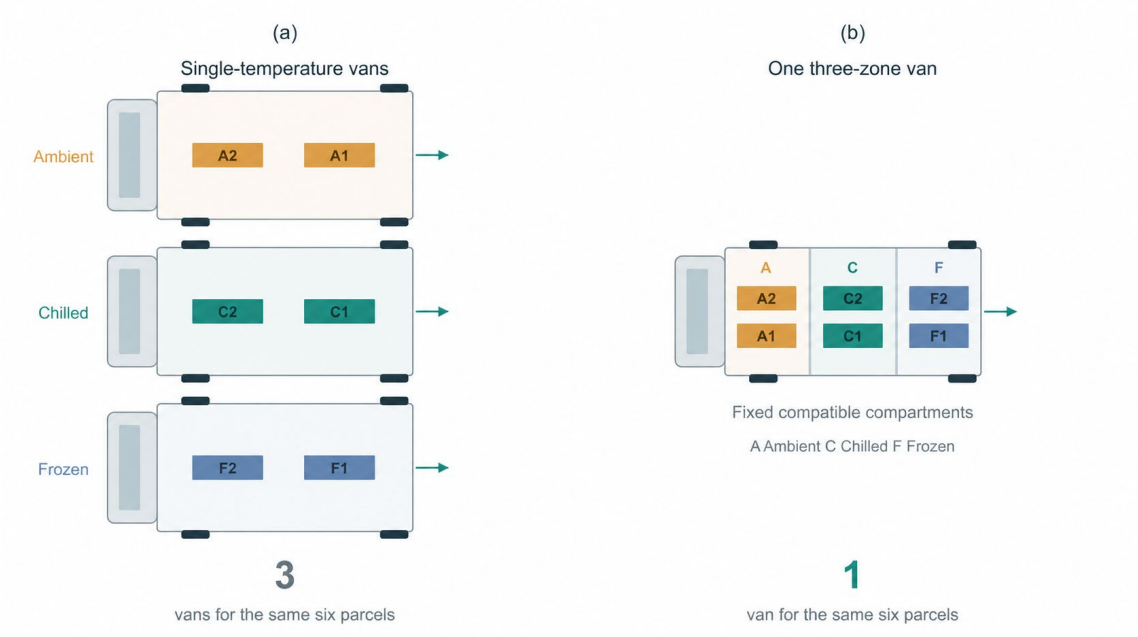


Figure 19 Single-temperature vehicles and a three-zone vehicle serving the same delivery demand.

Swap box loading operates at two linked levels. Parcels are arranged inside the selected module, and the complete loaded module is positioned on the larger vehicle and receiving van. A usable arrangement must also preserve mounting compatibility, access to the opening and the physical transfer path. A chassis-mounted body uses actual mounting positions, while a box transported inside a cargo compartment uses that compartment's geometry.

Routing proposes parcel-to-box assignment α_{ib} through (S1). For one loading check, $\overline{\alpha}_{ib}$ is that fixed candidate assignment, not an externally required manifest. Let $A_{ib,\mathrm{in}}$ contain admissible internal placements and binary $p_{iba,\mathrm{in}}$ select one. The internal placement condition is

$$\sum_{a \in A_{ib,\mathrm{in}}} p_{iba,\mathrm{in}} = \overline{\alpha}_{ib}. \qquad \text{(L1)}$$

Bounds, non-overlap, support and unloading conditions (14)–(22) are applied inside each box, together with its internal payload limit. Any fixed thermal zones follow the same compatibility and containment rules as (T1)–(T2). One internal arrangement is retained during whole-box transfer. The customer delivery ranks used for support chronology exclude transfer events.

New binary π_{bkat} selects the external position and orientation of box b on carrier k in load state t. Presence follows the routing candidate. A present box uses one valid position and keeps it while remaining on that carrier. The external choice is linked to the same internal packing. Non-overlap, support and access are checked in every physical state, including intermediate box transfers. Unplanned whole-box rearrangement during customer service is excluded, and unloading bound (22) remains global across boxes and loose parcels.

Axle feasibility is extended from departure and customer unloadings to every actual load state on both carriers. Let $\mathcal{K}$ contain the operating vans and larger vehicles, $\mathcal{G}_k$ its actual axles, and t a physical load state. Derived R_{gkt} is axle load and fixed $\overline{R}_{gk}$ its limit, both in mass-equivalent kg. The extension of (23)–(24) requires

$$0 \le R_{gkt} \le \overline{R}_{gk} \qquad (g \in \mathcal{G}_k, k \in \mathcal{K}, t). \tag{L3}$$

The calculation includes the empty vehicle, box tare, remaining contents and their positions. The larger vehicle uses its actual axle configuration and load-sharing model.

As illustrated in Figure 20, the Swap box extension checks loading feasibility throughout replenishment and vehicle transfers, supporting fewer depot returns while preserving space, weight and access requirements.

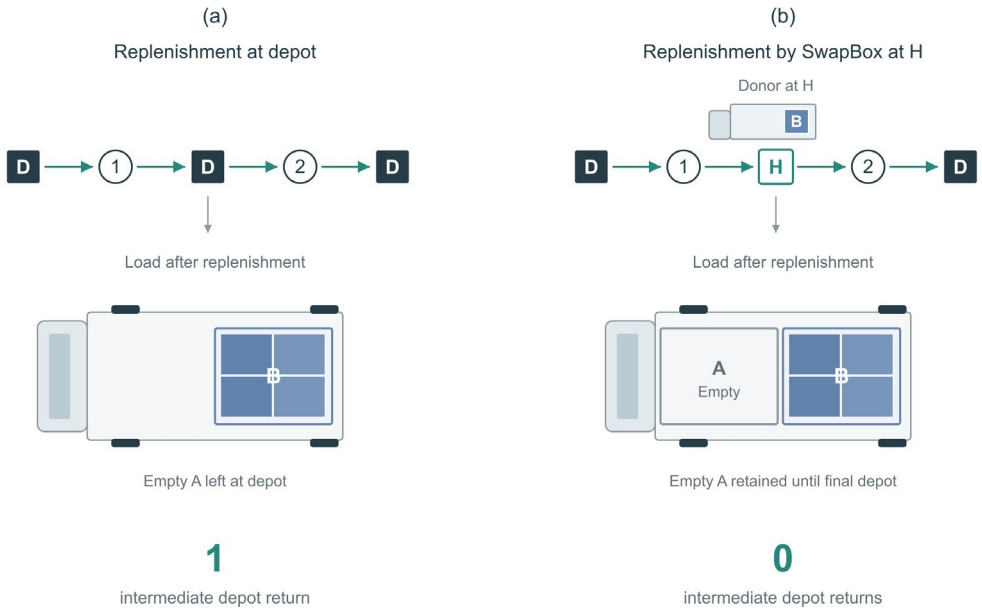


Figure 20 Comparison of depot replenishment and Swap box transfer at an intermediate location.

The complete-vehicle centre of gravity is reassessed after pickups and unloadings using the same mass accounting. The priorities in (12) are retained, with the terms in (13) adapted to the relevant states. Compactness concerns the contents inside boxes and loose cargo, rather than the fixed outside dimensions of a box.

Outputs identify parcel positions and orientations, selected zones and box contents, whole-box positions, unloading displacement, axle loads, complete-vehicle centre of gravity and utilisation. Figure 21 shows an arrangement from a saved run. The remaining payload at each stop is supplied for consumption assessment, while loading conflicts are returned to routing for reassignment or sequence repair. Detailed result fields are specified in Appendix C, Section C.2.

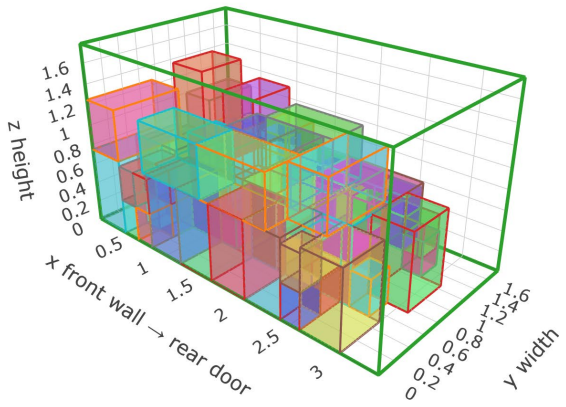


Figure 21. Computed front-wall loading arrangement sample with 66 parcels.

4.3.3 Charging Scheduling Engine

The Charging Scheduling Engine determines which charger each electric vehicle uses and when charging takes place before its next departure. It translates the energy requirements of the planned missions into a schedule that respects vehicle availability and shared depot resources. A vehicle is ready only when its required departure state of charge has been reached, and plug-out is complete before departure.

Each charging request represents one vehicle visit to one depot. The input specifies arrival and departure times, arrival state of charge, and the required departure state of charge. State of charge, or SoC, expresses the available battery charge as a percentage. Vehicle data provide usable battery capacity, accepted charging power, and compatible connectors. Depot data define charger types and quantities, rated power, efficiency, handling times, charging windows, and the site power limit. Electricity tariffs and objective priorities complete the scheduling configuration.

Figure 22 shows how these inputs are converted into feasible charging alternatives, charger assignments and a checked schedule. Detailed input and output fields are specified in Appendix C, Section C.3. The base scheduling formulation and variable glossary are provided in Appendix D, equations (25) – (41).

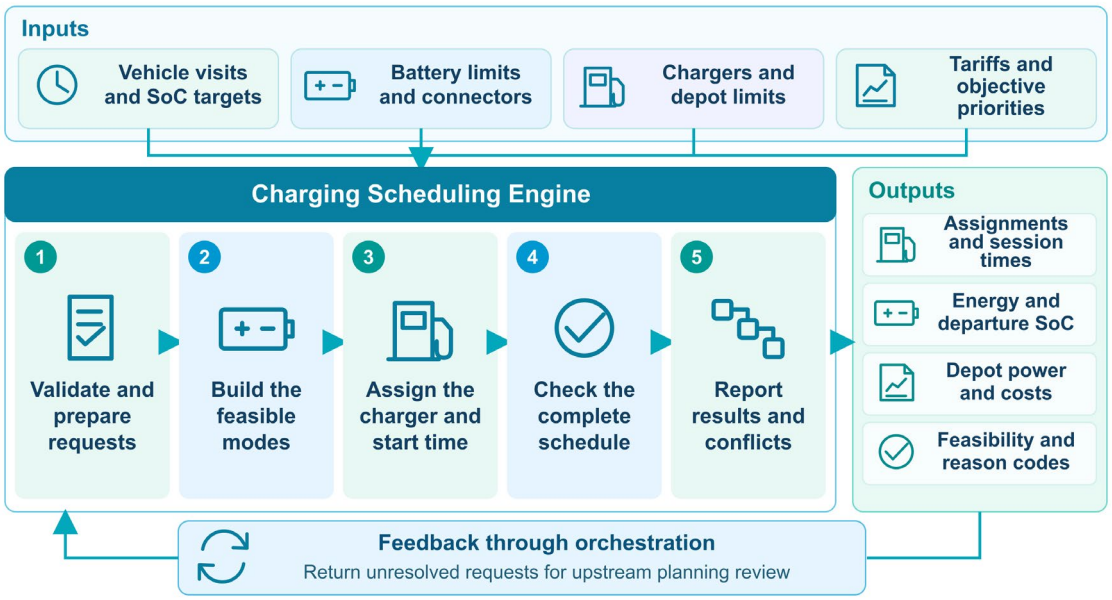


Figure 22. Charging workflow from vehicle energy requirements to a checked schedule and planning feedback.

For a request r requiring additional charge, fixed $C_{r,\mathrm{use}}$ is usable battery capacity in kWh. Fixed $\mathrm{SOC}_{r,\mathrm{arr}}$ and $\mathrm{SOC}_{r,\mathrm{req}}$ are the arrival and required departure percentages. The battery energy requirement $E_{r,\mathrm{req}}$ in kWh follows (25).

$$E_{r,\mathrm{req}} = C_{r,\mathrm{use}} \frac{\mathrm{SOC}_{r,\mathrm{req}} - \mathrm{SOC}_{r,\mathrm{arr}}}{100}. \tag{25}$$

A request already meeting its target requires no energy transfer. Otherwise, the engine constructs candidate charging modes using compatible charger classes and admissible starting intervals. Each mode contains a complete uninterrupted session, its duration and its grid-power profile. Charger and vehicle power limits, charging efficiency and any enabled SoC-dependent charging curve determine the energy delivered. Grid energy includes charging losses and is therefore distinguished from the energy added to the battery.

Let R denote the requests represented in the scheduling model and M_r the feasible modes of request r. Binary x_{rm} selects a mode, binary z_r indicates service, and continuous s_r is charging start in minutes from the planning origin. Equations (26)–(28) allow at most one mode per request and keep its start within its feasible interval. Charger-chain variables sequence sessions under (31)–(34). The number of simultaneously occupied chargers cannot exceed the available units of the relevant class.

Occupancy includes plug-in, energy transfer and plug-out. A later session on the same physical charger must wait until that occupancy ends. Charging windows and departure deadlines are embedded in the feasible modes. Start times are continuous unless an optional starting grid is supplied. Such a grid restricts the permissible starts without rounding charging duration or energy demand.

Different chargers may operate together only within the depot electrical allowance. Let L_m contain the constant-power segments of mode m, with fixed grid power $p_{m\ell}$ in kW for segment ℓ. Binary $u_{rm\ell b}$ indicates that the selected segment is active at relevant segment-start event b in event set B. Fixed $P_{\max}$ is the depot limit and non-negative e is any explicitly permitted exceedance, both in kW. Equation (35) requires

$$\sum_{r\in R}\sum_{m\in M_r}\sum_{\ell\in L_m} p_{m\ell}u_{rm\ell b} \le P_{\max} + e, \qquad b \in B. \tag{35}$$

For a hard depot limit, the exceedance is zero. A soft allowance is used only when explicitly configured, bounded and penalised. It does not increase charger or vehicle power ratings. The same segment representation supplies the peak-power measure in (36) and the time-dependent energy-cost calculation.

The first optimisation level maximises the number of served charging requests under (37). Later levels apply the configured objectives in priority order while preserving the attained objective values. Prioritising dead-time minimisation favours completion close to departure. This reduces the interval for which a vehicle waits after reaching its target SoC, rather than charging every vehicle as early as possible.

Non-negative δ_r measures dead time after plug-out and before departure for a served request, in minutes. Its model bounds are given in (29)–(30). When dead-time minimisation is active, the scheduler minimises the fleet total J_{dead} defined in (38).

$$J_{\text{dead}} = \sum_{r\in R} \delta_r. \tag{38}$$

This timing objective is an operational proxy, not a prediction of battery capacity loss. A separate degradation-cost objective in (40) uses battery replacement cost, assumed lifetime energy throughput, and a charging-rate multiplier when the required data are supplied. Calendar ageing, temperature-dependent ageing and cycle-count effects are not independently predicted by the present model. Electricity expenditure in (39) and peak demand in (41) can also be selected. Lower-priority cost savings cannot reduce the maximum number of served requests.

The scheduler uses a sparse mixed-integer linear programme in Python, solved with HiGHS through SciPy. Optimisation is performed separately for each depot. Compatibility and charging profiles are prepared before solving, while the model selects modes, start times and charger sequences. Afterwards, the schedule is checked against the original energy requirements, charging windows, occupancy and power limits. The corresponding core constraints and objective definitions are collected in Appendix D.

The output identifies each request, vehicle, depot and physical charger, together with charging and occupancy intervals, energy supplied, achieved departure SoC and dead time. Depot summaries report peak demand and remaining headroom. Costs are reported where defined, and statuses and reason codes distinguish served requests from unresolved ones. Figure 23 visualises the session intervals recorded in the supplied output JSON example. The example uses constant-power charging and leaves battery degradation cost undefined.

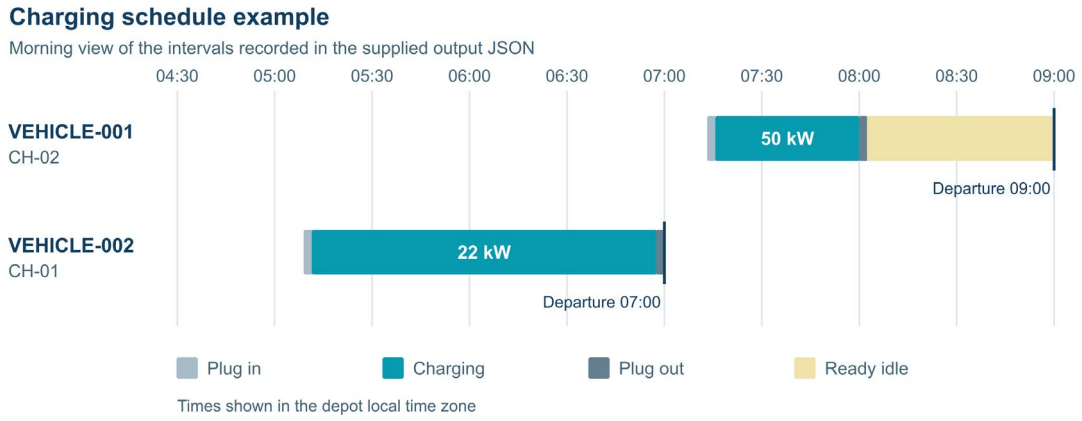


Figure 23. Charging and handling intervals recorded in the supplied output JSON example.

A fully feasible schedule meets every required charging request within the configured constraints. A partial schedule is not a feasible plan for the complete mission. The returned reasons can identify incompatible connectors, insufficient time, closed windows, or limited depot power. Failure to find a schedule before a solver time limit is distinguished from proven infeasibility of the configured model.

An unresolved request is returned to orchestration for review. Another admissible charger or charging time can be considered first. If readiness still cannot be achieved, an authorised change to vehicle availability, mission assignment or departure planning requires the affected routes, loading plans and energy requirements to be checked again. Required SoC and departure deadlines are not silently relaxed to make the result appear feasible.

Charging follows routing, cargo loading and consumption assessment. Geofencing and Swap box operations can change vehicle arrival and departure times, while thermal operation can change mission energy demand. Their effects reach charging through updated availability and SoC requirements. The scheduler returns readiness and limiting conditions but does not itself change parcel assignments, routes or energy predictions. Any upstream revision therefore reopens the relevant feasibility checks before the mission is accepted.

4.3.4 Vehicle Consumption Engine

The Vehicle Consumption Engine (VCE) provides the energy and battery-feasibility layer of the Shift2Zero (S2Z) analytical suite. **This Engine estimates the energy required by an electric Light Commercial Vehicle (eLCV) to execute a planned route and assesses whether the mission can be completed within the available battery energy.** It is designed for urban logistics operations, where frequent acceleration and deceleration, road gradients, vehicle payload, ambient conditions, auxiliary loads, and temperature-controlled cargo strongly influence energy demand. The engine further adapts vehicle-energy modelling to the project's mission-focused eLCV concepts and innovations, including modular and multi-temperature cargo bodies, thermal and energy-management solutions, regenerative braking, swappable cargo concepts, and dual transport of people and goods. It supports both pre-trip planning and operational monitoring. Before dispatch, it evaluates route feasibility, estimates expected energy consumption and identifies potential charging requirements. During operation, it can

incorporate observed vehicle states and telemetry to update the estimated State of Charge (SoC) and reassess the remaining mission. The engine therefore combines route characteristics, vehicle properties, cargo configuration, and environmental conditions to reconstruct the mission at segment level and estimate the resulting mechanical demand, electrical demand, and battery-state evolution. These elements form a continuous mission assessment, linking initial conditions to evolving energy demand and ultimately to whether the mission can be completed within the available battery reserve, as illustrated in Figure 24.

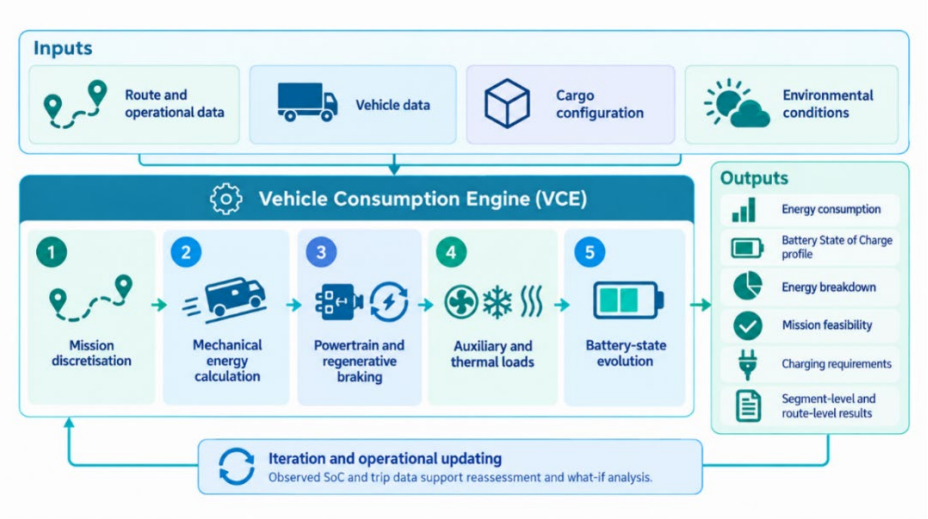


Figure 24: End-to-end computational pipeline

At the planning level, it determines whether a proposed route can be executed while maintaining a configurable battery reserve. It provides energy estimates that can subsequently support charging planning and operational decisions, while its detailed segment-level outputs enable analysis of where and why energy is consumed. The same framework can also be used for scenario and what-if analysis, allowing the effects of payload, driving behaviour, road conditions, ambient temperature, auxiliary demand and battery condition to be evaluated systematically.

The Engine interfaces with the VRP Engine, Charging Scheduling Engine, Cargo Loading Engine, and the Road Data Extraction Engine. The VRP Engine provides route, vehicle, and operational information, while the Cargo Loading Engine provides loading configuration and payload evolution. The Road Data Extraction Engine supplies route geometry and elevation data for segment-level energy calculation. The Vehicle Consumption Engine combines these inputs with the vehicle model and operational parameters to calculate energy demand, battery state, and route feasibility. A detailed specification of the engine's inputs and outputs is provided in Appendix C – Detailed Input and Output Specifications.

The VCE represents each vehicle mission as an ordered sequence of route segments and stops. The model combines four categories of input: operational and spatial information, vehicle characteristics, cargo configuration, and environmental conditions. Figure 25 shows how the mission is discretized into segments and, where required, shorter micro-segments for the energy calculation. Each segment contains the information required to reconstruct vehicle movement, including distance, speed, road

gradient or elevation change, stop behavior, ambient temperature, and traffic conditions. Vehicle inputs include mass, payload capacity, battery characteristics, motor and drivetrain parameters, regenerative braking limits, and auxiliary loads. Cargo operations are represented through loading and unloading information by cargo zone, allowing vehicle mass and thermal demand to evolve throughout the mission.

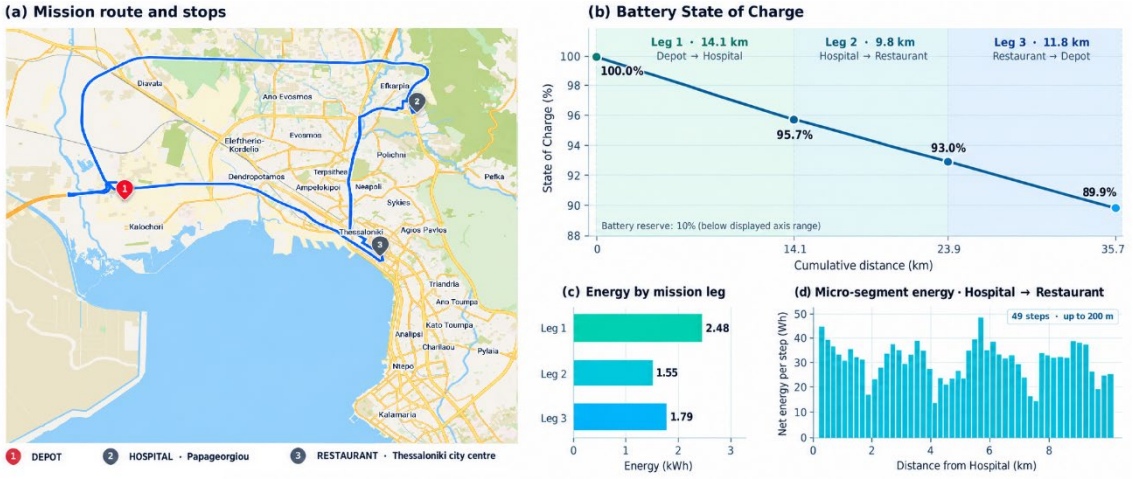


Figure 25: Mission discretisation

Having the mission discretised, the model then evaluates the mechanical demand associated with each simulated movement phase. For a vehicle of mass m, travelling at speed v on a road with slope angle θ, the principal road-load components are rolling resistance, aerodynamic drag and grade resistance. Together with the change in kinetic energy, these forces determine the mechanical energy required for each phase:

$$E_{\text{mech}} \approx \left(F_{\text{roll}} + F_{\text{drag}} + F_{\text{grade}}\right)\Delta s + \frac{1}{2}m\left(v_{\text{end}}^2 - v_{\text{start}}^2\right) \tag{42}$$

The route is simulated segment by segment and, where required, subdivided into shorter micro-segments. This allows energy demand to respond to payload, speed, road gradient and urban driving conditions rather than relying on a fixed consumption-per-kilometre factor. The mechanical demand is then converted into battery-side traction demand through the motor, transmission and inverter models. The model calculates net battery electrical power ($P_{\text{batt,net}}$) by combining traction with base electrical loads (P_{aux}), inverter standby losses, cabin HVAC (P_{HVAC}), battery thermal management (P_{BTM}), and cargo refrigeration (P_{ref}). For the S2Z pilot vehicle using eutectic cooling, the compressor power term is set to zero, while ventilation demand remains included. Depot pre-cooling and charging of the eutectic plates are accounted for separately from battery consumption during the route:

$$P_{\text{batt,net}} = P_{\text{batt,trac}} + P_{\text{aux}} + P_{\text{HVAC}} + P_{\text{BTM}} + P_{\text{ref}} - P_{\text{regen}} \tag{43}$$

Dwell periods are explicitly represented, allowing auxiliary consumption during stops to be calculated even when traction power is zero. The resulting electrical demand is passed to the battery model, which updates the vehicle state after each simulated phase while accounting for usable capacity, SoH, temperature effects, and battery operating

limits. The complete calculation therefore follows a consistent chain from route characteristics and vehicle properties to vehicle behavior, electrical demand, and the resulting SoC trajectory. The detailed mathematical formulations for these models are provided in Appendix D - Mathematical Formulations and Variable Glossary.

The model incorporates several extensions that are particularly relevant to the project's eLCV use cases. Cabin HVAC is represented using a heat-pump-based approach in which electrical demand depends on the difference between ambient and target cabin temperature and on the degradation of heat-pump performance under low-temperature conditions. Temperature-controlled cargo can be represented through multiple independent thermal zones, including frozen, refrigerated, and dry conditions. Heat entering each zone is estimated from the thermal characteristics of the cargo compartment and the ambient environment. Where eutectic or phase-change storage is available, this thermal storage is used before compressor energy is drawn from the battery. Depot pre-cooling and eutectic charging are treated separately from vehicle operation and are therefore accounted for as depot energy demand rather than as battery consumption during the route. Battery thermal management is also represented through a dedicated control mechanism that maintains the battery within its preferred operating temperature range. Together, these mechanisms allow the model to represent situations in which non-traction demand becomes a substantial component of total vehicle energy consumption.

Environmental and battery-condition effects are incorporated directly into the simulation rather than being represented through a single generic consumption correction. Cold-weather operation affects both battery usability and vehicle energy demand, while SoH influences available capacity and battery electrical behavior. Driver behavior can also be varied through configurable parameters controlling acceleration, deceleration, and overall driving style. At the end of each simulation step (Δt), the battery state updates continuously according to net energy exchanged:

$$\mathrm{SoC}_{t+\Delta t} = \mathrm{SoC}_t - \frac{P_{\mathrm{batt,net}} \Delta t}{E_{\mathrm{avail}}} \tag{44}$$

These capabilities allow the same route to be evaluated under different operating assumptions without modifying the underlying route definition.

Once all route segments have been simulated, the engine evaluates route feasibility using the minimum SoC reached anywhere along the mission, with the route considered feasible if and only if, for every segment t:

$$\min_t(\mathrm{SoC}_t) \geq \mathrm{SoC}_{\mathrm{reserve}} \tag{45}$$

This is important because considering only the final SoC could mask an earlier battery shortfall if regenerative braking subsequently increases the battery state. The model therefore identifies the point at which the reserve constraint is first violated and quantifies the associated energy shortfall. When a route is found to be infeasible, the engine generates remediation suggestions such as introducing an intermediate charging stop before the critical segment, reducing payload, or increasing the amount of pre-conditioning performed while the vehicle is connected to the depot power supply. These

results can then be passed to the charging and decision-support components for further optimization.

The engine also supports operation beyond the initial route-planning stage through real-time state estimation and model updating. A one-dimensional Kalman filter combines the modelled battery state with available telemetry measurements, including observed SoC and energy consumption, to reduce measurement noise and correct the estimated state during vehicle operation. The updated state is used to re-simulate the remaining segments and provide a revised assessment of energy requirements and route feasibility. In parallel, a telemetry-based calibration component uses historical vehicle trips to adjust selected parameters, including rolling resistance, aerodynamic drag and regenerative performance. This allows the model to progressively reflect observed vehicle characteristics, driver behavior and vehicle ageing. The detailed algorithms and workflows for these advanced modules are described in Appendix D - Mathematical Formulations and Variable Glossary.

The engine produces route-level and segment-level results, including energy consumption, regenerative energy, auxiliary demand, battery capacity, final and minimum SoC, consumption per kilometer, and route feasibility. Segment-level provenance allows results from micro-segments to be traced back to individual route legs. As illustrated in Figure 26, these outputs support downstream processes, including Charging Scheduling and Decision Analytics, while detailed RouteResult data enable operational monitoring, reporting and further analysis.



Figure 26:Outputs & Downstream Ecosystem

The VCE is implemented as a modular Python application, with separate modules for vehicle dynamics, drivetrain efficiency, regeneration, battery behaviour, thermal demand, feasibility assessment and telemetry-based updating. This structure allows the individual components to be updated or reused while maintaining defined interfaces with the other S2Z engines. A detailed description of the software architecture and core algorithms is provided in Appendix D - Mathematical Formulations and Variable Glossary.

To maintain computational efficiency and applicability to repeated route evaluation, several simplifications are deliberately retained. Acceleration and deceleration are

represented through quasi-steady phases rather than explicit jerk modelling, battery cell temperature is approximated by ambient temperature unless an alternative value is available, and driver behavior is represented through configurable parameters. These assumptions provide a practical balance between physical fidelity, computational cost and data availability for the project's eLCV applications.

Overall, the VCE extends conventional energy-consumption modelling to account for the specific vehicle technologies and operational configurations introduced in S2Z. Its unified, segment-level framework accommodates heterogeneous eLCVs, modular cargo bodies, Swap box operations and combined passenger–freight transport through vehicle characteristics and evolving mission conditions. It also incorporates multi-temperature cargo requirements, eutectic thermal storage and energy-management technologies through dedicated physical and auxiliary energy models, while accounting for the effects of geofencing through upstream routing inputs.

4.4 Analytics-Simulation and What-If Scenario Analysis Block

4.4.1 Decision Analytics Engine

The Decision Analytics Engine connects operational planning, performance monitoring and project evaluation. It processes operational data into indicators and analytical results that support logistics operators and cities in assessing urban logistics performance. Within Shift2Zero, it supports the evaluation of vehicle concepts and their suitability for different missions. Its algorithms and data structures are aligned with the WP7 impact assessment framework and the data requirements established through T5.1 and T5.2. The implemented analytical functions process outputs from the operational engines alongside company mission records and available vehicle data. These inputs include route distances, stop sequences, delivery outcomes, service times, cargo utilization, energy consumption and charging records. The data structures also accommodate observations from vehicle communication systems, supporting comparison between planned and actual operations.

KPI calculations operate at mission, vehicle, fleet, or pilot level, according to the indicator definition. For example, kilometers per completed delivery relates to the total distance traveled to the number of completed deliveries within the same assessment period. Relevant depot, positioning, and return movements remain included in the calculation. Each result retains its calculation basis, distinguishing planned, modelled, and observed values. Operational analytics cover distance travelled, delivery productivity, route duration, waiting time, service reliability, and vehicle utilization. The functions also process empty running, transshipment handling and charging-related downtime where the required records are available. Energy analytics relate consumption to distance, completed deliveries, payload, and operating conditions. Environmental calculations combine the operational records with the applicable factors and assessment boundaries defined under WP7. Economic calculations use the same operational quantities together with supplied tariffs and cost parameters. Each calculation retains its assumptions and source data, allowing results to be reviewed and reused in the corresponding impact assessment.

The engine examines both mission feasibility and the performance of accepted plans. Feasibility results identify whether the evaluated routing, loading, energy, and

charging requirements are satisfied. Performance indicators describe how vehicles, cargo space, time, and infrastructure are used. The analytical functions support the identification of inefficiencies within otherwise feasible operations. These include long positioning journeys, low cargo utilization, excessive waiting, and limited charging flexibility. Route and stop records allow such effects to be located and related to their operating conditions. **Where observed execution data are available, comparisons also identify recurring delays and differences between model assumptions and actual performance.**

Table 5: Indicative Decision Analytics Outputs, Calculations and Visualisations

Analytic	Calculation or result	Indicative presentation
Trip completion	Completed trips (count) and completed / scheduled trips (%).	Daily or weekly trend and trip-status bars.
Order fulfilment	Fully fulfilled orders (count) and fulfilled / requested orders (%).	Order-status bars and fulfilment trend.
On-time delivery	Deliveries completed within the agreed window / completed deliveries (%).	Daily trend and lateness distribution.
Trip and fleet distance	Distance per trip and fleet total (km), including positioning and return legs.	Trip-distance histogram and daily fleet trend.
Distance per delivery	Total vehicle distance / completed deliveries (km/delivery).	Daily trend and comparison by vehicle or mission.
Trip duration	Elapsed time from departure to return (min/trip).	Duration histogram or box plots by mission profile.
Delivery productivity	Completed deliveries / vehicle duty hours (deliveries/h).	Daily trend and vehicle productivity distribution.
Waiting and handling	Waiting, delivery service and transfer handling time (min/stop or trip).	Time-breakdown bars and stop-duration distributions.
Cargo utilisation	Occupied volume or payload / usable capacity (%), evaluated along the route.	Route load profiles and utilisation distributions.
Trip energy consumption	Measured or modelled energy per trip and fleet total (kWh).	Trip-energy histogram and daily fleet trend.
Energy intensity	Energy / distance (kWh/km) and energy / completed deliveries (kWh/delivery).	Intensity trends and energy-versus-distance scatter.
Operational greenhouse gas emissions	Energy or fuel use with WP7 emission factors (kg CO_2e/trip and g CO_2e/km).	Emissions time series and baseline comparisons.
Operational cost	Energy, labour and configured operating costs (EUR/trip and EUR/delivery).	Cost-breakdown bars and cost-per-delivery trend.
Charging performance	Requests reaching required departure SoC (%), charger utilisation (%) and depot peak power (kW).	Charging Gantt chart and depot power time series.
Innovation and baseline comparison	Absolute KPIs and relative change (%) under comparable demand and service requirements.	Paired comparison plots and KPI comparison tables.

The reporting functions organize calculated indicators with the information required for their interpretation. This includes the assessment period, mission profile, vehicle configuration, data coverage, and comparison baseline. Results can be examined for individual missions or aggregated across vehicles, operating periods, and pilot

scenarios. Their presentation through operator and city interfaces forms part of the platform integration work under T5.4. The analytical outputs also support comparisons involving S2Z innovations. Examples include delivery productivity associated with swap-box operations, utilization of multi-temperature cargo space, and the operational effects of access restrictions. Comparisons use consistent demand and service requirements, while retaining any differences that affect interpretation. This provides a basis for assessing the contribution of an innovation under the conditions represented by the available data. Table 5 presents an indicative selection of outputs from the Decision Analytics Engine, including their calculation basis and suitable statistical and graphical presentations.

WP7, particularly T7.1, provides the methodological basis for indicator selection and impact assessment. The implemented analytics follow the relevant KPI definitions and calculation requirements established through coordination with WP7. Their data requirements are aligned with the work performed under T5.1 and T5.2. The operational data structures retain identifiers linking scenarios, vehicles, routes and stops. Calculated indicators can therefore be traced back to their contributing records and processing steps. This supports consistent calculation and review across pilot and simulation activities. Updates to the assessment framework can be incorporated through the corresponding indicator definitions and calculation parameters.

The Decision Analytics Engine is implemented at algorithm level through KPI calculation functions, structured data processing, and internal analytical pipelines. The operational engines retain the quantities and identifiers needed to extract, aggregate, and trace their results. This establishes the computational basis for operational analysis and its contribution to the WP7 assessment framework. The remaining work concerns the integration of these functions into the S2Z Urban Logistics Management Platform under T5.4. It includes connecting the relevant data streams, configuring the pilot-specific indicators and exposing monitoring and reporting functions through the platform interfaces. The final configuration is aligned with pilot objectives and data availability. Algorithmic implementation is therefore established, while platform deployment and operational use remain part of the integration activities.

4.4.2 Simulation Engine

The Simulation Engine is an auxiliary platform module for strategic scenario assessment. It supports the examination of large-scale adoption of Shift2Zero solutions by logistics operators and municipalities. Detailed simulation methods, scenario development and results are addressed under T5.5. As illustrated in Figure 27, the

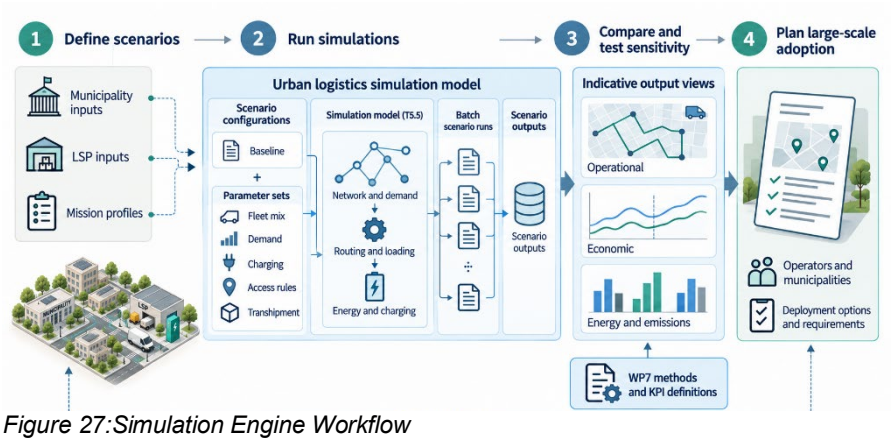


Figure 27:Simulation Engine Workflow

proposed module follows a predefined input/output structure. Municipality inputs describe urban access conditions, infrastructure and policy settings. Logistics service provider inputs describe demand, fleet characteristics and operating arrangements. Mission profiles define the delivery patterns and service requirements represented in each scenario.

A reference scenario establishes the comparison baseline. Alternative configurations vary the adoption of S2Z vehicle concepts, fleet composition, demand, charging provision or transhipment arrangements. Each configuration retains its assumptions and parameter values so that differences between scenarios can be interpreted. Simulation runs evaluate the selected configurations using the relevant operational, optimisation and consumption models. Sensitivity analysis examines how changes in individual parameters or their combinations affect the results. **This identifies conditions that constrain deployment, infrastructure requirements and trade-offs between operating configurations.** The analysis supports questions such as how increased eLCV adoption affects fleet requirements, how charging provision limits service capacity and how transhipment changes delivery performance.

Outputs comprise scenario-level performance indicators, comparisons against the baseline and sensitivity results. The indicators follow the definitions and assessment methodology established under WP7. They cover operational performance, energy use, environmental effects and economic outcomes. Relevant safety indicators are included where supported by the available data and assessment models.

These results provide inputs to the project impact assessment. They also help operators and municipalities examine deployment options and identify the resources and operating conditions required for wider adoption. Each result retains its scenario configuration and modelling assumptions. Detailed development, calibration and scenario evaluation belong to T5.5. The platform integration provides access to scenario configuration, execution and comparison as an additional strategic planning function. **The common input/output structure connects simulation results with the Decision Analytics Engine and WP7 evaluation activities.**

4.5 Cross-Engine Dependencies and Interaction Logic

Operational planning produces a consistent route, loading plan, energy profile and charging schedule. Each engine evaluates quantities established by the preceding engines. A change in vehicle assignment or stop sequence can therefore invalidate several results. The three feedback loops address these dependencies by revising the candidate mission and repeating the affected assessments.

The process begins with company orders and the applicable mission profile. Municipal conditions add geofencing, access rules and service periods. Order Data Communication structures the demand. Road Data Extraction supplies distance and travel-time information, route geometry and available elevation profiles. Vehicle Communication provides available operating observations. The platform supplies vehicle characteristics and other reference records. The orchestrator prepares the requests and manages the exchanges throughout planning.

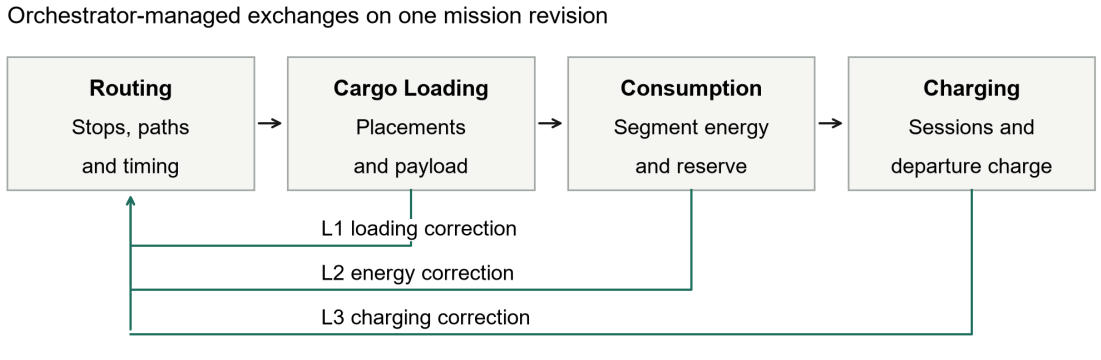


Figure 28: Operational Engine Dependencies and Orchestrator-Managed Revision Loops

Figure 28 illustrates the operational dependencies and the three orchestrator-managed revision loops. The Routing Engine assigns deliveries to vehicles and defines the stop sequence and timing. Road Data Extraction associates the ordered legs with paths and available terrain information. Cargo Loading uses the sequence to determine physical placement and unloading access. Consumption uses the resulting paths, payload history, and vehicle configuration to estimate energy at each segment. Charging then uses the mission requirement and vehicle availability to determine a feasible charging plan.

A feedback result identifies the affected route, physical units, or energy requirement. Routing evaluates permitted corrections, such as resequencing, relocating a delivery, or exchanging deliveries between routes. The orchestrator applies the selected revision and refreshes its dependencies. These are candidate-search operations within the coordinated workflow. A diagnostic alone does not establish that an automatic correction has been applied.

L1 Routing and Cargo Loading

L1 links the route assignments with the physical loading assessment. The orchestrator sends Cargo Loading the selected vehicle configuration, parcel or box dimensions, handling actions, and ordered stops. Cargo Loading checks fit and unloading access, then returns placements or conflicts. In Figure 29, six smaller parcels fit V1, but unit 4 is 120 cm wide and exceeds its 100 cm cargo width in the required orientation. The conflict is returned to the orchestrator, which requests a revised routing solution. Routing exchanges deliveries 4 and 6, assigning unit 4 to V2, which has a cargo width of 140 cm, and the 80 cm unit 6 to V1. The figure shows the original conflict and the revised vehicle allocations.

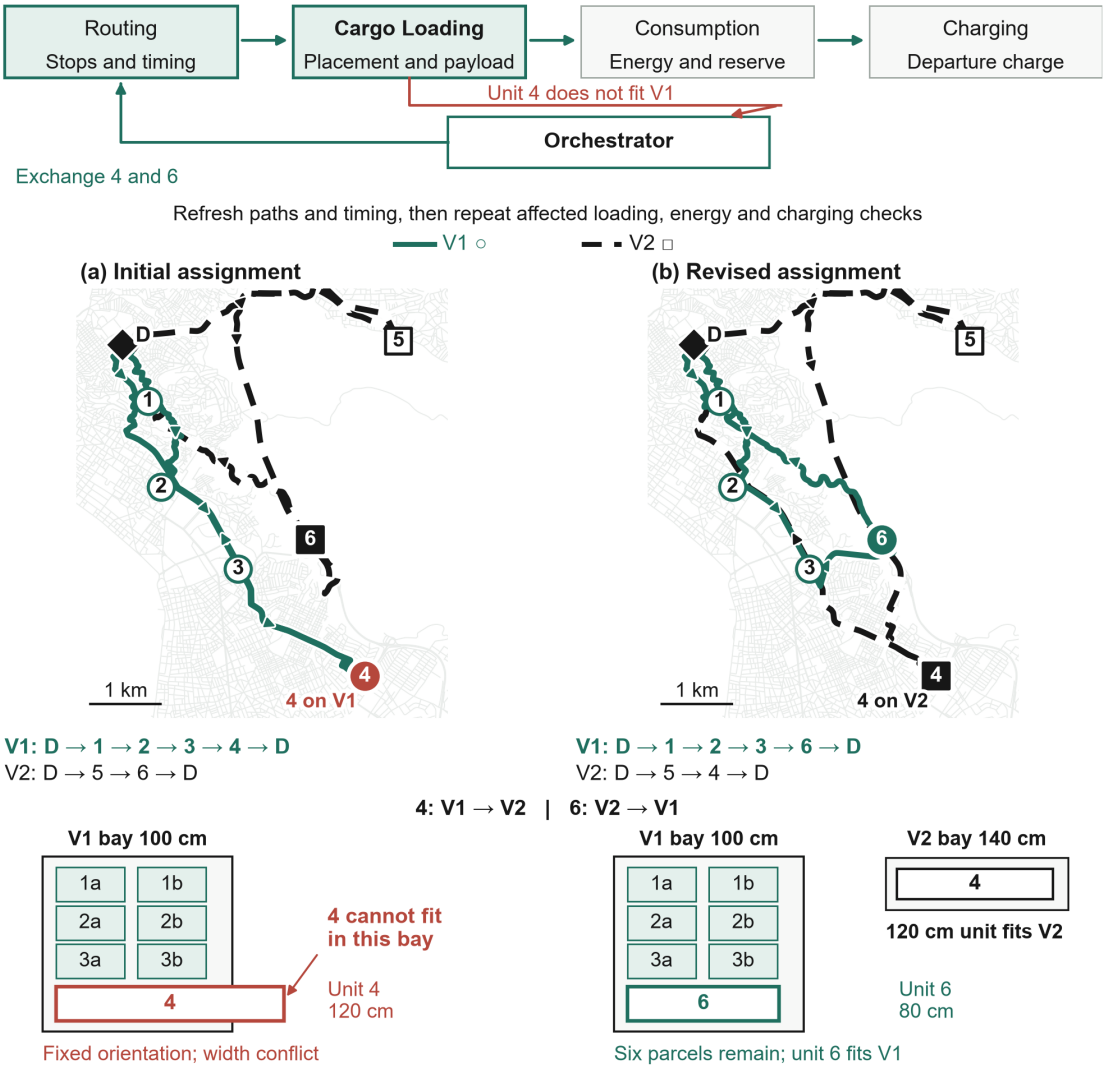


Figure 29: L1 example. Cargo-loading conflict and revised vehicle assignments. Roads: © OpenStreetMap contributors.

Routing exchanges deliveries 4 and 6, placing unit 4 in V2, whose cargo width is 140 cm, and the 80 cm unit 6 in V1. Road Data Extraction refreshes the paths and timing for both sequences. Cargo Loading checks both revised arrangements and their unloading access. Consumption then uses the revised payload histories, and Charging reassesses

the resulting departure requirements. A different stop order can address an unloading conflict, while a fit conflict can require reassignment or a different cargo configuration. The orchestrator repeats the affected assessments until the required checks pass or the configured search limit is reached.

Check-only mode retains the supplied mission. Coupled operation uses the conflict to guide a new routing candidate. Without dimensions, geometry is marked as not tested; mass, volume, and temperature compatibility remain active. A required geometric assessment remains unresolved until dimensions are supplied.

L2 Routing and Energy Consumption

L2 links the current route with its battery trajectory. Consumption receives the road paths, available elevation, vehicle parameters, changing payload, departure charge and reserve requirement. Driving conditions, auxiliaries and refrigeration parameters complete the assessment. It returns segment energy and the points where battery reserve is violated. Figure 30 illustrates an energy-related routing revision. V1 and V2 initially have 16 kWh and 24 kWh of battery energy, respectively. The original mission causes V1 to fall below its required 3 kWh reserve. The orchestrator returns this diagnosis to Routing, which reassigns delivery 4 from V1 to V2. Following reassessment, the revised missions finish with 4 kWh and 13 kWh, respectively, satisfying the reserve requirement for both vehicles.

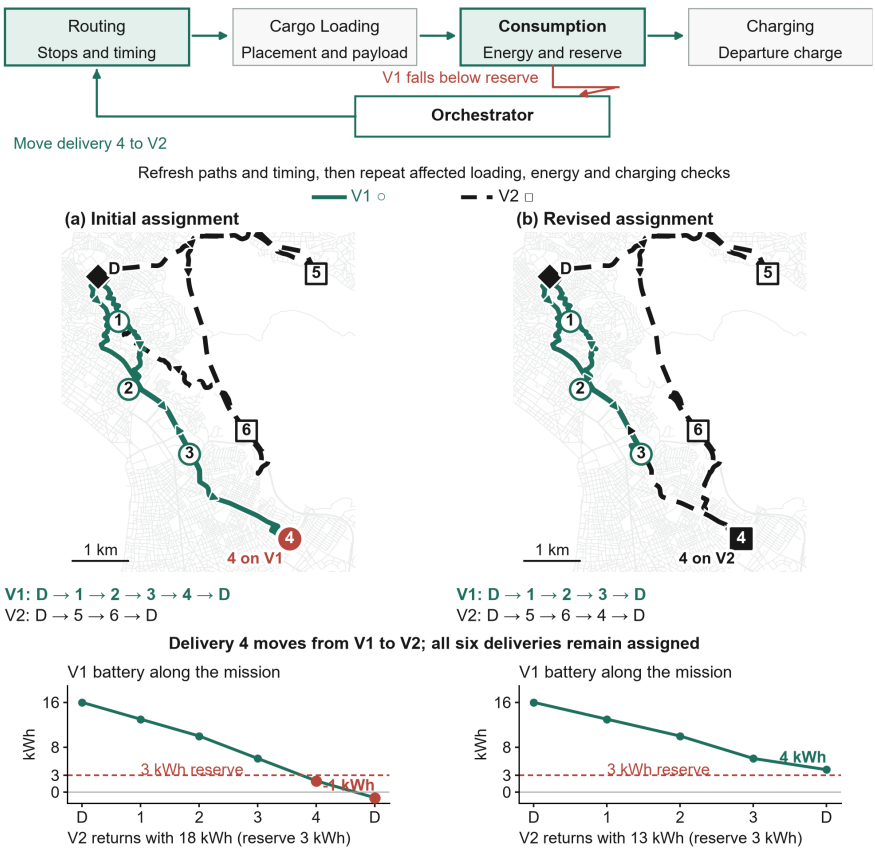


Figure 30: L2 example. Energy-related routing revision and battery-feasibility assessment. Roads: © OpenStreetMap contributors.

Road Data Extraction supplies the revised legs, travel times and available elevation profiles. Cargo Loading rechecks the new assignments and stop sequences when geometry is available. Consumption recalculates both vehicles with their revised paths and payload histories; the receiving vehicle must also satisfy reserve. Charging then checks the departure state needed by each mission. Permitted candidates can also change the stop order, exchange deliveries or substitute a vehicle. Removing a visit replaces its adjacent road legs, so its energy effect cannot be found by subtracting that visit's distance alone.

L3 Routing and Charging Scheduling

L3 tests whether the departure state required by Consumption can be achieved before the mission starts. The orchestrator supplies Charging with the battery state at arrival, vehicle availability, ready-by time, and required departure charge. Compatible connectors, charging curves, vehicle limits, charger occupancy, plug handling, and depot power constrain the sessions. Charging returns the schedule, achievable state, and any shortfall.

Figure 31 illustrates a charging infeasibility and its resolution through route reassignment. V1 initially has 12 kWh of battery energy, while its planned mission requires 17 kWh plus a 3 kWh reserve. This creates an 8 kWh charging requirement. However, the available one-hour charging window at 6 kW and 90% efficiency can supply only 5.4 kWh, leaving a 2.6 kWh shortfall. The orchestrator therefore requests a routing revision. Reassigning delivery 4 to V2 reduces V1's required charging energy to 3 kWh, which can be supplied within the available charging window.

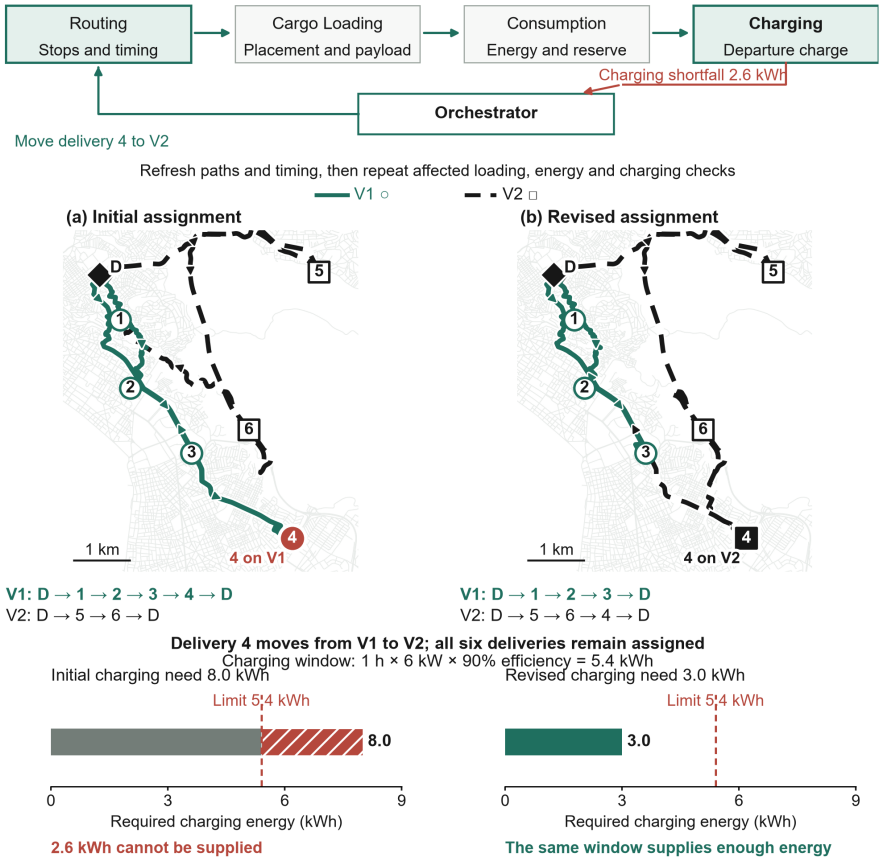


Figure 31: L3 example. Charging infeasibility and revised delivery assignment. Roads: © OpenStreetMap contributors.

The orchestrator first distinguishes a charging adjustment from a mission change. Another admissible session or compatible charger can be tested while the mission remains fixed. If the available resources still cannot meet readiness, Routing evaluates a permitted reassignment, exchange, additional vehicle, or departure strategy. Charging reports the limiting condition and does not alter delivery assignments itself.

Here Routing moves delivery 4 to V2, which starts with 24 kWh and can cover its revised 11 kWh mission plus the 3 kWh reserve. Road Data Extraction updates both routes, Cargo Loading checks the changed allocation, and Consumption recalculates both battery trajectories. V1's revised mission requires 12 kWh, so 3 kWh of charging now meets its departure target. Charging evaluates that requirement within the same one-hour window. The service remains assigned, and the complete revision is checked again.

Shared constraints and mission revision

The loops evaluate one shared mission revision. L1 changes payload and stop sequence for L2. L2 can change the vehicle or delivery allocation and reopen L1. L3 can reveal an unattainable departure requirement and reopen both. Each successful check remains valid only while its contributing inputs remain unchanged.

Table 6. Principal dependencies refreshed after an applied mission change.

Applied change	Affected assessments refreshed, as applicable
Stop order or road path	Leg geometry and terrain, timing and city rules, unloading access, payload history and segment energy.
Delivery or box reassignment	Both vehicle routes, physical fit, temperature compatibility, box custody, transfer timing, energy and charging requirements.
Vehicle or cargo configuration	Eligibility, capacity and geometry, temperature layout, vehicle energy model and charger compatibility.
Departure state or availability	Energy trajectory, mission timing, customer and geofence periods, shared transfer appointments and charging readiness.

Swap boxes retain their contents, external dimensions and tare. Paired donor and receiver actions preserve receipt before delivery and a common transfer interval. Empty Swap boxes retain their tare mass and are assumed to be folded, with their carriage imposing no binding cargo-space constraint during the remaining route. Multi-temperature operation adds the selected layout and each unit's compatible zone. Reassignment must preserve custody and temperature compatibility.

The receiving route also needs assessment. Additional cargo can increase traction, service time, and auxiliary energy. Changed assignments or sequences reopen geometric loading when dimensions are available. Swap-box moves preserve contents and custody. Refrigeration follows the selected temperature configuration and can consume energy during driving, waiting, and service. Depot thermal preparation remains separate from on-route battery demand.

Battery reserve must hold throughout the mission. Efficiency objectives compare feasible candidates. Current observed charge differs from required departure charge: charging can support a later mission if the required state is achievable. A changed departure state returns to Consumption for reassessment. Route changes also reopen geofence eligibility, service intervals, and supported road restrictions.

Vehicle substitution changes cargo geometry, capacity, temperature capability, energy parameters and charging compatibility. A changed departure can affect customer windows, a swap-box appointment or a geofence service period, reopening L1 and L2 as needed. Alternative charging locations must be supplied, compatible and within scheduling scope. Cost objectives select feasible options without overriding readiness or mission requirements.

Requests and results retain mission, route, vehicle, stop and physical-unit identifiers. Paths, cargo states, energy profiles and charging sessions refer to the same revision. The orchestrator records the cause of each change and invalidates affected results. A proposed move remains distinct from an applied revision, preserving traceability across engines.

Parcel priorities guide the permitted search, with mandatory service protected. Reassignment and resequencing precede any authorised partial-service exception. Results identify unserved deliveries and their causes. Exhausting a bounded search does not establish that no feasible plan exists. Conflicts remain explicit as reserve targets and readiness deadlines are not relaxed to conceal them. Integrated computational

performance, including engine synchronisation and orchestration, will be assessed under T5.4, with current development directed towards offline planning on the day before operation and calculation limits to be defined according to the available server hardware and computing resources.

External LSP and municipal operations execute accepted plans during pilot trials. New orders, delays and vehicle observations can reopen remaining work, retaining completed deliveries, current position, cargo custody and observed battery state. Execution records support the implemented Decision Analytics and KPI pipelines, whose integration continues under T5.4. The proposed T5.5 Simulation module uses these dependencies for strategic assessment and WP7 impact evaluation. Pilot-specific activation of the coordinated workflow remains under T5.4.

5. Conclusions and Extension Roadmap

This chapter concludes Deliverable D5.3 by summarizing the outcomes of Task 5.3 against the objectives defined in the Grant Agreement and outlining the next steps towards platform integration and pilot and large-scale simulation activities. It first discusses advanced operational features, technical assessment, and further development needs. It then summarizes the analytical capabilities developed and their achievements, assesses integration, readiness, scalability and transferability across different mission profiles, and identifies the remaining activities required to support subsequent S2Z tasks.

5.1 Integrated Operational Features and Advanced Extensions

The S2Z Analytical Suite can be extended to connect mission planning with city infrastructure and daily fleet operations. The following extensions build on the routing, loading, consumption, charging, and communication engines described in section 4. They would use the shared mission representation and orchestration logic to assess changes consistently across the affected engines.

Geofencing could become an automated interface between municipalities and logistics service providers for managing access and curbside use. City systems would publish spatial boundaries, vehicle eligibility, operating periods and loading-space conditions in machine-readable form. The platform would translate these updates into routing and stopping constraints. In the opposite direction, LSPs could communicate expected arrivals, requested stopping durations and actual arrival and departure events. Where municipal services support booking, the interface could request loading slots and incorporate confirmations into the mission schedule. The municipality would retain responsibility for allocating public space and defining its conditions of use.

Connection with urban infrastructure could also supply the operational information needed to update these missions. In Thessaloniki, integration with the Thessaloniki Smart Mobility Living Lab could provide observed and predicted traffic conditions to the Road Data Extraction Engine. Updated road travel times would feed routing, ETAs and consumption assessment. Infrastructure events affecting an active mission could trigger reassessment of its remaining stops, using the vehicle's current position, cargo and battery state.

The same interfaces could support the freight component of city digital twins. City models would provide network, infrastructure and scenario conditions, while the S2Z tools would return freight-demand representations, vehicle routes, delivery-stop activity, energy requirements and performance indicators. This exchange would require consistent location identifiers, time references, and definitions of the simulated operations. *During S2Z, the Bologna city case will explore this connection to support the freight aspect of the city's digital twin as a potential extension of project continuation activities*. The intended contribution is the provision of freight models and operational outputs that can be used within the municipal modelling environment.

Routing could apply delivery priorities throughout mission execution, with continuously updated ETAs, deviation detection and alerts for threatened service windows. Replanning could change the sequence or vehicle assignment of remaining deliveries

according to their priorities, while retaining completed stops and cargo custody. Each proposed revision would undergo the relevant access, loading, energy and charging checks before becoming the active plan. Historical execution records could also improve customer-specific estimates of parking search, unloading, and service duration. These estimates could distinguish delivery location, time of day, shipment characteristics and observed operating conditions, replacing fixed stop durations where sufficient observations are available.

Reverse flows would extend the mission model to include collections, product returns, reusable packaging, and empty cargo units alongside deliveries. Load and available capacity would be updated after each collection or delivery. Further transshipment configurations could include Urban Consolidation Centers, microhubs and cross-docking facilities. Planning would represent cargo availability, hub opening periods, handling time and capacity, and the synchronization of incoming and outgoing vehicles. Joint scheduling of feeder and last-mile legs would extend the current Swap box configuration in which the donor route and meeting periods are fixed.

LSP–driver communication would maintain a common operational plan through route updates, acknowledgements and execution reports. Drivers could manually adjust a proposed stop sequence or access point, with the resulting mission checked against the applicable constraints. Dispatchers would receive delivery status, delays and reported deviations, while drivers would receive the accepted route revision and updated instructions. Mission and revision identifiers would distinguish proposed changes from the plan currently in force and keep vehicle observations associated with the correct route.

Charging could be extended beyond depot scheduling to include opportunity charging at publicly accessible locations. External data would describe charger location, connector compatibility, available power, operating conditions, tariffs and current availability. The Routing, Vehicle Consumption and Charging Scheduling Engines would jointly assess whether a charging stop fits the mission, accounting for access, detour, expected waiting, charging duration and subsequent delivery commitments. Charging during suitable stops could be evaluated where the infrastructure and operating conditions allow it. The revised battery trajectory would need to satisfy the energy requirements and reserve for the remaining mission.

Figure 32 illustrates how updated city curbside information can trigger a revision of an ongoing delivery route. At 09:10, the loading bay at customer 1 is reported unavailable until 09:35. In panel (a), retaining the original stop sequence requires 17 minutes of waiting, delaying arrival at priority customer 2 until 09:49, beyond its 09:35 deadline. In panel (b), the toolbox changes the sequence to serve customer 2 first, at 09:22, and reach customer 1 after the bay reopens. The revised plan also includes an optional public charging stop after the deliveries, subject to charger availability, compatibility and sufficient time; the displayed depot return includes this stop. The lower diagram shows the associated information exchange: city services provide bay availability, the toolbox communicates the revised route and ETAs to the LSP and driver, and operational status and confirmation return to the toolbox. Arrival and dwell-time estimates can also inform city curbside management.

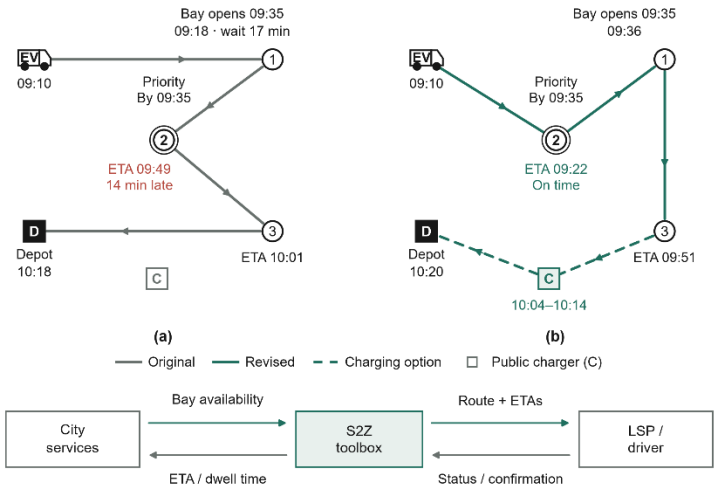


Figure 32: Curbside-triggered route replanning with optional charging and operational data exchange.

5.2 Summary of Deliverable's Achievements

D5.3 establishes the S2Z Analytical Suite as a multipurpose and customizable set of optimization, modelling, communication, simulation, and decision-support components for eLCV operations. The suite brings together routing, cargo loading, energy consumption, charging, vehicle and operational data, and analytical functions within a common workflow. This provides the technical basis for applying the same analytical capabilities to different vehicle configurations, operating conditions, and mission profiles.

The main achievement of Task 5.3 is the development of the analytical components and their interaction logic. Common data representations and defined interfaces allow information to be exchanged between the engines, while the modular structure allows individual models and optimization functions to be configured or extended without redesigning the complete workflow. This supports the proposal objective of developing a multipurpose suite that can be adapted to different eLCV deployment scenarios.

The developed workflow supports the main operational decisions required for efficient eLCV deployment. Routing determines vehicle and mission assignments, cargo loading evaluates physical loading feasibility, vehicle consumption assesses energy demand and battery feasibility, and charging scheduling evaluates charging requirements. These functions can exchange results and constraints so that routing, loading, energy, and charging are considered as interdependent aspects of the same mission.

The suite also incorporates the vehicle-specific and mission-specific characteristics introduced by S2Z, including heterogeneous eLCVs, modular and multi-temperature cargo, thermal and energy-management requirements, regenerative braking, dual transport, geofencing, and transshipment-related configurations. This provides a basis for extending the analytical workflow as additional vehicle concepts, operational configurations, or regulatory requirements emerge.

Preliminary testing and validation activities provide initial evidence of the functional behavior and interoperability of the developed components. The deliverable also presents representative computational examples illustrating the functionality of individual engines and their interactions, including the handling of routing, loading, energy and charging constraints. These examples support the technical assessment of the developed components and provide reference configurations for subsequent pilot validation.

The achievements of Task 5.3 are assessed against the commitments defined in the Grant Agreement, as summarized in Table 7. The assessment distinguishes between objectives addressed through the developed analytical components and those for which the technical foundation has been established but further integration or validation is required under subsequent project activities.

Table 7:Assessment of Task 5.3 Achievements Against Grant Agreement Commitments

Task 5.3 commitment	Status	Evidence in D5.3
Develop a multipurpose and customisable analytical suite	Addressed	Nine interconnected analytical engines with configurable models, parameters, and operational constraints
Develop optimisation capabilities including vehicle and dynamic routing and charging	Addressed	VRP, loading, charging, and energy-related optimisation and assessment functions
Seamlessly integrate analytical data streams	Addressed	Implemented data structures, interfaces and orchestration services supporting inter- engine feedback mechanisms, coordinated execution and exchange of analytical results.
Support end-users in the efficient deployment of new eLCVs	Addressed	Mission-level routing, loading, energy, charging, and decision-support functions applicable to pilot configurations.
Support scalability and transferability across mission profiles	Addressed	Parameterised vehicle, cargo, charging, routing, environmental, and regulatory representations
Support IoT-enabled operational safety and autonomy forecasting	Foundation established	Vehicle communication and telemetry interfaces, vehicle-state information, energy estimation, and monitoring capabilities

The results therefore address the core technical scope of T5.3 while establishing the interfaces and analytical foundation required for the subsequent platform and pilot activities. Completion of the full operational environment, connection to pilot data

streams, and large-scale deployment remain part of the subsequent development sequence.

5.3 Integration Readiness

The analytical suite has established the technical basis required for integration into the S2Z Urban Logistics Management Platform. The engines use defined inputs and outputs and shared data structures for the exchange of orders, vehicles, routes, cargo, energy, battery, and charging information. The resulting interfaces allow the individual components to operate as a connected workflow rather than as standalone analytical tools.

The main dependencies between the engines have also been established. Routing results can be used by the Cargo Loading and Vehicle Consumption Engines, loading results can update payload and energy requirements, and energy assessments can provide inputs to Charging Scheduling. This provides the integration logic required to combine routing, loading, energy, and charging decisions within a common mission workflow.

The current implementation therefore provides an integration-ready analytical foundation for T5.4. Further work will connect the engines to the platform services and pilot data sources, complete the orchestration of the full workflow, and validate the integrated system under operational conditions. The modular interfaces also allow individual engines, data sources, and analytical functions to be extended or replaced as new S2Z requirements emerge during the project and in potential future applications.

5.4 Scalability and Transferability

Scalability and transferability are central objectives of T5.3. The analytical suite is not tied to a single vehicle or mission configuration. Instead, vehicle, cargo, battery, charging, routing, environmental, and regulatory characteristics are represented through configurable data and model parameters.

This approach allows the engines to be adapted to different eLCV platforms, fleet compositions, cargo types, depot configurations, and urban operating conditions. It also supports future mission profiles that were not explicitly represented during development by extending the relevant data structures, constraints, or decision logic rather than replacing the complete analytical framework.

The same principle applies beyond the project. The modular interfaces and parameterized models provide a basis for adapting the engines to future urban logistics requirements, including new vehicle configurations, changing access regulations, new charging concepts, additional data sources, and more complex multimodal or transshipment-based operations. This addresses the T5.3 objective of supporting the scalability and transferability of the analytical methods across mission profiles and provides potential for reuse after the project.

5.5 Next Steps towards T5.4 and T5.5

The work completed in T5.3 provides the analytical components, data structures, and integration logic required for the subsequent activities under T5.4 and T5.5.

Under **T5.4, Integration of tools & data streams for enhanced decision-making**, the developed engines will be brought together within the S2Z Urban Logistics Management Platform. The next steps include organizing the corresponding data sources within the platform, completing the interfaces and orchestration required for end-to-end workflows, and providing user access to the available analytical tools through a common interface. The platform will subsequently be connected to the pilot sites to support the management, monitoring, and optimization of their operations and to assess the performance of the integrated system.

Under **T5.5, Scenario simulations for pilot optimization**, the analytical and simulation capabilities will be used to replicate pilot operations, support the configuration of pilot activities, and assess alternative operational scenarios. The simulation models will extend the pilot configurations to larger-scale instances and evaluate the scalability and potential of the S2Z solutions under different policy, business, and operational conditions. The simulation outputs from T5.5 will inform the preparation and configuration of the demonstrations under T6.1 and support their implementation under T6.2. Moreover, the resulting analyses will provide quantitative inputs to the impact assessment activities in WP7 and support the evaluation of the deployment potential of the developed solutions.

References

Basso, R., Kulcsár, B., Egardt, B., Lindroth, P., & Sanchez-Diaz, I. (2019). Energy consumption estimation integrated into the electric vehicle routing problem. *Transportation Research Part D: Transport and Environment*, *69*, 141–167. https://doi.org/10.1016/j.trd.2019.01.006

Bruglieri, M., Çatay, B., Keskin, M., Mancini, S., & Pisacane, O. (2025). The mixed-fleet vehicle routing problem with low emission zones. *Transportation Research Part E: Logistics and Transportation Review*, *201*, Article 104230. https://doi.org/10.1016/j.tre.2025.104230

Chen, L., Liu, Y., & Langevin, A. (2019). A multi-compartment vehicle routing problem in cold-chain distribution. *Computers & Operations Research*, *111*, 58–66. https://doi.org/10.1016/j.cor.2019.06.001

Chen, W., Zhang, D., Van Woensel, T., Xu, G., & Guo, J. (2023). Green vehicle routing using mixed fleets for cold chain distribution. *Expert Systems with Applications*, *233*, Article 120979. https://doi.org/10.1016/j.eswa.2023.120979

Chi, J., Song, R., & He, S. (2026). Capacitated vehicle routing problems with three-dimensional loading constraints: A systematic review and new taxonomy. *IET Intelligent Transport Systems*, *20*(1), Article e70281. https://doi.org/10.1049/itr2.70281

Cieślik, W., & Antczak, W. (2023). Research of load impact on energy consumption in an electric delivery vehicle based on real driving conditions: Guidance for electrification of light-duty vehicle fleet. *Energies*, *16*(2), Article 775. https://doi.org/10.3390/en16020775

Di Martino, A., Miraftabzadeh, S. M., & Longo, M. (2022). Strategies for the modelisation of electric vehicle energy consumption: A review. *Energies*, *15*(21), Article 8115. https://doi.org/10.3390/en15218115

Docker Inc. (2026, May 28). Compose file reference. Docker Documentation. https://docs.docker.com/reference/compose-file/

Ferreira, K. M., de Queiroz, T. A., Munari, P., & Toledo, F. M. B. (2024). A variable neighborhood search for the green vehicle routing problem with two-dimensional loading constraints and split delivery. *European Journal of Operational Research*, *316*(2), 597–616. https://doi.org/10.1016/j.ejor.2024.01.049

Fiori, C., & Marzano, V. (2018). Modelling energy consumption of electric freight vehicles in urban pickup/delivery operations: Analysis and estimation on a real-world dataset. *Transportation Research Part D: Transport and Environment*, *65*, 658–673. https://doi.org/10.1016/j.trd.2018.09.020

Froger, A., Mendoza, J. E., Jabali, O., & Laporte, G. (2019). Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. *Computers & Operations Research*, *104*, 256–294. https://doi.org/10.1016/j.cor.2018.12.013

Gao, Y., Zhang, S., Zhang, Z., & Zhao, Q. (2024). The stochastic share-a-ride problem with electric vehicles and customer priorities. *Computers & Operations Research*, *164*, Article 106550. https://doi.org/10.1016/j.cor.2024.106550

Co-funded by the European Union

The Shift2Zero project has received funding from the European Union Horizon Europe Programme: project num. 101192375. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

GitLab. (n.d.-a). CI/CD pipelines. GitLab Documentation. https://docs.gitlab.com/ci/pipelines/

GitLab. (n.d.-b). Get started with GitLab CI/CD. GitLab Documentation. https://docs.gitlab.com/ci/

Goeke, D., & Schneider, M. (2015). Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, *245*(1), 81–99. https://doi.org/10.1016/j.ejor.2015.01.049

gRPC Authors. (n.d.). Introduction to gRPC. gRPC Documentation. https://grpc.io/docs/what-is-grpc/introduction/

Hangfire. (n.d.). Hangfire documentation. https://docs.hangfire.io/en/latest/

He, Z., Shen, X., Sun, Y., Zhao, S., Fan, B., & Pan, C. (2021). State-of-health estimation based on real data of electric vehicles concerning user behavior. *Journal of Energy Storage*, *41*, Article 102867. https://doi.org/10.1016/j.est.2021.102867

Hiermann, G., Puchinger, J., Ropke, S., & Hartl, R. F. (2016). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, *252*(3), 995–1018. https://doi.org/10.1016/j.ejor.2016.01.038

Huang, Y., Zhao, L., Van Woensel, T., & Gross, J.-P. (2017). Time-dependent vehicle routing problem with path flexibility. *Transportation Research Part B: Methodological*, *95*, 169–195. https://doi.org/10.1016/j.trb.2016.10.013

Huber, S., & Geiger, M. J. (2017). Order matters – A variable neighborhood search for the swap-body vehicle routing problem. *European Journal of Operational Research*, *263*(2), 419–445. https://doi.org/10.1016/j.ejor.2017.04.046

Keycloak. (n.d.). Keycloak documentation. https://www.keycloak.org/documentation

Krebs, C., Ehmke, J. F., & Koch, H. (2021). Advanced loading constraints for 3D vehicle routing problems. *OR Spectrum*, *43*(4), 835–875. https://doi.org/10.1007/s00291-021-00645-w

Kucukoglu, I., Dewil, R., & Cattrysse, D. (2021). The electric vehicle routing problem and its variations: A literature review. *Computers & Industrial Engineering*, *161*, Article 107650. https://doi.org/10.1016/j.cie.2021.107650

Łebkowski, A. (2024). Energy consumption optimization for an electric delivery vehicle. *Energies*, *17*(22), Article 5665. https://doi.org/10.3390/en17225665

Ma, H., Yang, R., & Li, X. (2026). Delivery routing for a mixed fleet of conventional and electric vehicles with road restrictions. *International Journal of Production Research*, *64*(2), 677–700. https://doi.org/10.1080/00207543.2024.2324054

Mahyari, E., & Freeman, N. (2025). Electric vehicle fleet charging management: An approximate dynamic programming policy. *European Journal of Operational Research*, *327*(1), 263–279. https://doi.org/10.1016/j.ejor.2025.04.031

Masmoudi, M. A., Baldacci, R., Fontaine, P., & Khayya, E. (2026). A novel multi-compartment vehicle routing problem with time windows: Reconfigurable vehicles and dynamic compartments. *European Journal of Operational Research*, *335*(3), 787–801. https://doi.org/10.1016/j.ejor.2026.03.033

Microsoft. (n.d.-a). Health checks in ASP.NET Core. Microsoft Learn. https://learn.microsoft.com/en-us/aspnet/core/host-and-deploy/health-checks

Microsoft. (n.d.-b). Migrations overview. Microsoft Learn. https://learn.microsoft.com/en-us/ef/core/managing-schemas/migrations/

Microsoft. (n.d.-c). Overview of ASP.NET Core SignalR. Microsoft Learn. https://learn.microsoft.com/en-us/aspnet/core/signalr/introduction

Miri, I., Fotouhi, A., & Ewin, N. (2021). Electric vehicle energy consumption modelling and estimation—A case study. *International Journal of Energy Research*, *45*, 501–520. https://doi.org/10.1002/er.5700

Muñuzuri, J., Grosso, R., Cortés, P., & Guadix, J. (2013). Estimating the extra costs imposed on delivery vehicles using access time windows in a city. *Computers, Environment and Urban Systems*, *41*, 262–275. https://doi.org/10.1016/j.compenvurbsys.2012.05.005

Osorio-Castañeda, C. D., Orejuela-Cabrera, J. P., & Bravo-Bastidas, J. J. (2025). Refrigerated electric vehicle routing considering time-dependent temperature energy consumption models. *Annals of Operations Research*, *349*(3), 1817–1854. https://doi.org/10.1007/s10479-025-06579-8

Ostermeier, M., Henke, T., Hübner, A., & Wäscher, G. (2021). Multi-compartment vehicle routing problems: State-of-the-art, modeling framework and future directions. *European Journal of Operational Research*, *292*(3), 799–817. https://doi.org/10.1016/j.ejor.2020.11.009

Ostermeier, M., Martins, S., Amorim, P., & Hübner, A. (2018). Loading constraints for a multi-compartment vehicle routing problem. *OR Spectrum*, *40*(4), 997–1027. https://doi.org/10.1007/s00291-018-0524-4

Parragh, S. N., & Cordeau, J.-F. (2017). Branch-and-price and adaptive large neighborhood search for the truck and trailer routing problem with time windows. *Computers & Operations Research*, *83*, 28–44. https://doi.org/10.1016/j.cor.2017.01.020

Pelletier, S., Jabali, O., & Laporte, G. (2018). Charge scheduling for electric freight vehicles. *Transportation Research Part B: Methodological*, *115*, 246–269. https://doi.org/10.1016/j.trb.2018.07.010

Pelletier, S., Jabali, O., Laporte, G., & Veneroni, M. (2017). Battery degradation and behaviour for electric vehicles: Review and numerical analyses of several models. *Transportation Research Part B: Methodological*, *103*, 158–187. https://doi.org/10.1016/j.trb.2017.01.020

Pollaris, H., Braekers, K., Caris, A., Janssens, G. K., & Limbourg, S. (2015). Vehicle routing problems with loading constraints: State-of-the-art and future directions. *OR Spectrum*, *37*(2), 297–330. https://doi.org/10.1007/s00291-014-0386-3

Rastani, S., Yüksel, T., & Çatay, B. (2019). Effects of ambient temperature on the route planning of electric freight vehicles. *Transportation Research Part D: Transport and Environment*, *74*, 124–141. https://doi.org/10.1016/j.trd.2019.07.025

Shao, S., Guan, W., & Bi, J. (2018). Electric vehicle-routing problem with charging demands and energy consumption. *IET Intelligent Transport Systems*, *12*(3), 202–212. https://doi.org/10.1049/iet-its.2017.0008

Shi, J., Zeng, T., & Moura, S. (2023). Electric fleet charging management considering battery degradation and nonlinear charging profile. *Energy*, *283*, Article 129094. https://doi.org/10.1016/j.energy.2023.129094

Todosijević, R., Hanafi, S., Urošević, D., Jarboui, B., & Gendron, B. (2017). A general variable neighborhood search for the swap-body vehicle routing problem.

Computers & Operations Research, *78*, 468–479. https://doi.org/10.1016/j.cor.2016.01.016

Wang, X., Liang, Y., Tang, X., & Jiang, X. (2023). A multi-compartment electric vehicle routing problem with time windows and temperature and humidity settings for perishable product delivery. *Expert Systems with Applications*, *233*, Article 120974. https://doi.org/10.1016/j.eswa.2023.120974

Wu, J., Su, H., Meng, J., & Lin, M. (2023). Electric vehicle charging scheduling considering infrastructure constraints. *Energy*, *278*, Article 127806. https://doi.org/10.1016/j.energy.2023.127806

Yang, D., & Hyland, M. F. (2024). Electric vehicles in urban delivery fleets: How far can they go?. *Transportation Research Part D: Transport and Environment*, *129*, Article 104127. https://doi.org/10.1016/j.trd.2024.104127

Zhao, J., Zeng, Z., & Liu, Y. (2025). Electric vehicle routing problem considering traffic conditions and real-time loads. *Transportation Research Part C: Emerging Technologies*, *176*, Article 105150. https://doi.org/10.1016/j.trc.2025.105150

Appendix A - Overview of the Reviewed Literature

Table 8: Summary of the reviewed literature and its relevance to the S2Z Analytical Suite

Reference	Relevant S2Z Domain	Key Contribution & Focus for Shift2Zero
Kucukoglu et al. (2021)	VRP	Review of 136 EVRP studies by objective, energy model and constraints.
Hiermann et al. (2016)	VRP	Fleet size and mix of electric vehicle types; matching right-sized platforms to missions.
Goeke & Schneider (2015)	VRP	Mixed electric/conventional fleet with speed-, gradient- and load-dependent energy.
Yang & Hyland (2024)	VRP	Minimum-cost urban delivery fleet mix of electric and conventional vehicles.
Froger et al. (2019)	VRP / charging	Nonlinear charging functions inside the EVRP.
Basso et al. (2019)	VRP / energy	Road-network energy estimation (regeneration, mass) integrated into routing.
Shao et al. (2018)	VRP / energy	Consumption as a function of speed and cargo load in the EVRP.
Zhao et al. (2025)	VRP / energy	Energy-minimising EVRP with time-dependent speed and real-time load.
Gao et al. (2024)	VRP (people + freight)	Electric fleet serving prioritised passengers and parcels; dual-use eLCVs.
Huang et al. (2017)	VRP / road network	Path choice by departure time and congestion; road-network input to routing.
Pollaris et al. (2015)	Cargo loading	Review of vehicle routing with loading constraints.
Chi et al. (2026)	Cargo loading	Review and taxonomy of routing with 3D loading constraints.
Krebs et al. (2021)	Cargo loading	Load bearing, stability, axle-weight and reachability constraints for vans.
Ferreira et al. (2024)	Cargo loading / energy	2D loading with weight-dependent emissions; loading plan linked to energy.
Ostermeier et al. (2021)	Multi-temperature	Common framework for multi-compartment routing problems.
Ostermeier et al. (2018)	Multi-temperature	Loading of flexible compartments from the rear; multi-zone body layout.
L. Chen et al. (2019)	Multi-temperature	Cold-chain compartments with fuel used to keep temperature while waiting.
Masmoudi et al. (2026)	Multi-temperature / modular	Reconfigurable compartments of modular frigoboxes optimised with routes.
Wang et al. (2023)	Multi-temperature / EV	Multi-compartment EV routing with temperature and climate y settings.
W. Chen et al. (2023)	Multi-temperature / EV	Mixed electric/conventional cold-chain fleet with realistic energy model.
Huber & Geiger (2017)	Swap bodies	Swap-body routing with body exchange locations.
Todosijević et al. (2017)	Swap bodies / access	Swap-body routing with per-customer restrictions on vehicle combinations.
Parragh & Cordeau (2017)	Swap bodies / access	Truck-and-trailer routing; customers reachable only without trailer.
Pelletier et al. (2018)	Charging	Depot charging of electric freight fleets with prices, degradation and grid limits.
Wu et al. (2023)	Charging	Time-of-use charging with limited charging points and station power.
Shi et al. (2023)	Charging	CC/CV charging profile and degradation cost in fleet charging.

Mahyari & Freeman (2025)	Charging / IoT	Real-time depot charging under uncertainty with multi-connector chargers.
Pelletier et al. (2017)	Charging / battery	Tractable battery degradation and charge/discharge models for freight EVs.
Di Martino et al. (2022)	Energy consumption	Review of physics-based, microsimulation and data-driven consumption models.
Miri et al. (2021)	Energy consumption	Forward physics-based EV model; structure of the Consumption Engine.
Fiori & Marzano (2018)	Energy consumption / telemetry	Power-based model calibrated on 144 real urban freight EV trips.
Cieślik & Antczak (2023)	Energy consumption / telemetry	Measured payload effect on an electric delivery vehicle.
Rastani et al. (2019)	Energy consumption	Ambient temperature effect on fleet size, energy and routes.
Osorio-Castañeda et al. (2025)	Energy consumption	Refrigeration ≈44% of the energy of an electric refrigerated vehicle.
Łebkowski (2024)	Energy consumption	Predictive motor and gear control from route data; vehicle control strategies.
He et al. (2021)	Telemetry / battery	State of health from real vehicle operating data; usable capacity.
Muñuzuri et al. (2013)	Urban access	Routing with access time windows; extra vehicles and tour length.
Bruglieri et al. (2025)	Urban access	Mixed-fleet routing through low emission zones with tolls and charging.
Ma et al. (2026)	Urban access	Accessible time windows and road weight limits for a mixed fleet.

Appendix B- Details of technical implications

Table 9:Technical implications of S2Z innovations for the analytical engine modules

S2Z Innovation	VRP	Cargo Loading	Charging Scheduling	Vehicle Consumption	Road / Communication	Analytics & Simulation
Energy Management	Energy feasibility constraints and route reassessment	Payload effects on energy feasibility	Charging needs linked to route energy demand and SoC	Physics based consumption, SoC and auxiliary loads	Road, elevation, speed and vehicle-state inputs	Energy, charging and fleet-performance analysis
Swap box and Modular Cargo	Route sequence linked to payload changes	3D loading, Swap box fit and configuration	Charging demand reflecting dynamic payload	Dynamic mass by delivery leg	Cargo status and configuration updates where available	Configuration, capacity and energy comparison
Multi-Temperature Cargo	Thermal compatibility and duration constraints	Multi-zone loading and door-opening rules	Peak power including auxiliary loads and pre-cooling	Multi-zone thermal, HVAC and eutectic demand	Live zone temperature and door status	Separate propulsion and thermal KPIs
Geofencing and Access Constraints	LEZ, restricted-area and loading-access constraints	Vehicle and cargo fit for restricted operations	Charging linked to access and mission constraints where relevant	Energy effects of constrained urban routes	Geofences, road restrictions and vehicle position	Access, routing and operational scenario analysis
Tyre, Brake & Control Strategies	Elevation, gradient and braking-aware cost	Axle-load distribution for wear and braking	Reduced grid demand through recuperation	Regenerative braking, friction and 2D motor maps	Elevation, gradient and speed profiles	Non-exhaust emissions and energy gains

Appendix C – Detailed Input and Output Specifications

This appendix describes the main inputs and outputs of the Vehicle Routing Problem, Cargo Loading, Charging Scheduling and Vehicle Consumption Engines. The specifications are based on the available engine interfaces and example data structures. They identify the information required to execute the analytical functions, the results returned by each engine and the data exchanged with other components of the S2Z Analytical Suite.

C.1 Vehicle Routing Problem Engine

The Vehicle Routing Problem (VRP) Engine uses delivery demand, vehicle availability, depot locations and road-network information to determine vehicle assignments and delivery sequences. The planning process takes account of vehicle capacity, customer time windows, working hours and route-duration constraints. The VRP–3DST interface also contains the vehicle and parcel information required for subsequent loading assessment.

Table 10:VRP Engine input requirements

Input category	Required information
Planning period and depot	Service date, depot identifier and geographical coordinates, working-day start and end times, maximum route duration (min) and customer service time (min).
Fleet availability	Vehicle type identifiers, number and identifiers of available vehicles, maximum payload (kg) and internal cargo compartment dimensions (m).
Customer and delivery data	Customer identifiers, addresses and coordinates, delivery time windows, assigned parcels, parcel priorities, weights, dimensions and handling requirements.
Road-network information	Road-based distances and travel times between depot and customer locations, together with the geographical information required to construct the routes.

Road-network information is provided through the Road Data Extraction Engine, which interfaces with routing services such as OSRM and Valhalla. In the available VRP–3DST input example, customer and depot locations are specified through geographical coordinates rather than explicit distance and travel-time matrices. The corresponding output identifies OSRM as the routing information provider.

The vehicle and parcel characteristics are also used by the Cargo Loading Engine. Detailed loading constraints, including axle weights, parcel placement and unloading accessibility, are described in Section C.2.

Table 11:VRP Engine outputs

Output category	Reported information
Optimisation status	Solution status, execution time (s), overall feasibility and identifiers of unassigned customers or parcels.
Vehicle assignment	Number and identifiers of active vehicles, selected vehicle types and the customers and parcels assigned to each vehicle.

Co-funded by the European Union

The Shift2Zero project has received funding from the European Union Horizon Europe Programme: project num. 101192375. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

Route definition	Ordered customer sequence, delivery stops, associated parcel identifiers and road-path geometry represented by geographical coordinates.
Route performance	Total and vehicle-level distance (km), travel time, customer service time and route duration (min).
Capacity utilisation	Assigned payload (kg), payload utilisation, cargo volume (m^3) and volume utilisation.

The feasibility results indicate whether mandatory customer service, fleet availability and applicable route constraints have been satisfied. Customers or parcels that cannot be assigned are reported separately.

The calculated routes provide the delivery sequences and geographical paths required for subsequent energy assessment. Vehicle and parcel assignments are also passed to the loading assessment to verify whether the proposed deliveries can be physically accommodated.

C.2 Cargo Loading Engine

The Cargo Loading Engine assesses whether the parcels assigned to a vehicle can be accommodated within its available cargo space. In addition to vehicle payload and compartment dimensions, the assessment considers cargo weight distribution, axle limits, parcel orientation, stacking and access to parcels during unloading.

The engine uses the parcel assignments and customer sequence produced by the routing process. Vehicle geometry and axle characteristics are required to evaluate loading conditions that cannot be represented through aggregate weight and volume constraints alone.

Table 12:Cargo Loading Engine input requirements

Input category	Required information
Vehicle and cargo compartment	Vehicle identifier and type, internal cargo compartment length, width and height (m), and maximum payload (kg).
Axle configuration and weight distribution	Wheelbase (m), cargo front-wall offset from the front axle (m), empty front and rear axle loads (kg), and maximum permitted axle loads (kg).
Parcel characteristics	Parcel and customer identifiers, cargo type, length, width and height (m), weight (kg), priority, fragility and stackability.
Delivery and unloading sequence	Assigned customer sequence and parcel-to-customer relationships used to evaluate unloading accessibility at each stop.

The axle-related inputs are used to determine how cargo placement affects vehicle weight distribution. Parcel dimensions, orientation and stacking restrictions define the physical loading conditions, while the delivery sequence is used to assess whether parcels remain accessible when required.

The combined VRP–3DST output also includes an unloading movement limit.

Table 13:Cargo Loading Engine outputs

Output category	Reported information

Three-dimensional placement	Parcel and customer identifiers, cargo type, placement coordinates (m), dimensions after orientation and identifiers of supporting parcels, where applicable.
Centre of gravity	Overall longitudinal and lateral centre-of-gravity coordinates, position relative to the front axle, empty-vehicle centre of gravity and cargo centre of gravity (m).
Axle loading	Calculated front and rear axle loads, permitted axle-load limits and total loaded cargo weight (kg).
Capacity utilisation	Payload utilisation, total cargo volume (m³) and volume utilisation.
Unloading accessibility	Evaluation scope, maximum permitted parcel movements, maximum movements required by the loading arrangement and number of stops exceeding the limit.
Loading feasibility	Individual checks for parcel placement, orientation, compartment boundaries, overlap, support, stability, stackability, payload capacity, axle limits, unloading accessibility and route-duration compliance.

Parcel placement is represented in a three-dimensional coordinate system using the cargo compartment front wall, left side and floor as reference surfaces. Each parcel is associated with its assigned customer, allowing the loading arrangement to be assessed against the planned delivery sequence.

The output includes separate Boolean feasibility checks for the applicable physical and operational constraints. These cover whether all assigned parcels have been loaded, whether their orientation and placement are permitted, whether sufficient support and stability are maintained, and whether vehicle capacity and unloading requirements are satisfied.

Payload and volume utilisation are also reported through the route-performance results of the combined VRP–3DST interface. Together with the placement and axle-loading information, these results allow the routing and loading configuration to be assessed as part of the same delivery plan.

C.3 Charging Scheduling Engine

The Charging Scheduling Engine determines charging assignments and schedules using vehicle availability, battery requirements, depot infrastructure and electricity tariffs. The model takes account of when vehicles arrive and must depart, the energy required for their next mission, charger compatibility and the available electrical capacity of the depot.

The input structure also allows different optimisation objectives and optional battery-related costs to be represented. The active objectives and modelling assumptions depend on the configuration supplied to the engine.

Table 14:Charging Scheduling Engine input requirements

Input category	Required information
Planning and optimisation	Start and end timestamps of the planning horizon, optional time-step resolution, selected objectives, priorities and objective weights.
Charging constraints	Connector compatibility, permitted charging windows, required departure SoC (%), departure deadlines and depot power limit (kW).

Depot infrastructure	Depot identifier and local time zone, maximum depot power (kW), charger identifiers and quantities, rated power (kW), charging efficiency coefficient, connector types and plug-in/plug-out times (min).
Vehicle and battery characteristics	Vehicle identifier, nominal and usable battery capacity (kWh), maximum accepted charging power (kW), supported connectors and optional battery ageing parameters.
Charging requests	Vehicle and depot identifiers, arrival and required departure timestamps, arrival SoC (%) and required departure SoC (%).
Electricity tariffs	Tariff identifier and currency, time-dependent electricity prices, applicable day types, standing charge, VAT, network fees and soft-constraint penalties.
Optional battery and charging models	SoC-dependent charging power curves, charging-rate degradation penalties, battery cycle count, replacement cost, lifetime energy throughput and calendar-ageing parameters.

The engine supports four defined optimisation objectives: minimising non-charging dead time, electricity costs, battery degradation costs and depot peak power demand. In the example configuration, dead-time and electricity-cost minimisation are active, with weights of 0.6 and 0.4 respectively.

Charging feasibility is evaluated against vehicle availability, departure requirements, connector compatibility and depot power limits. Constraints may be configured as mandatory requirements or, where supported, as soft constraints with permitted exceedances and associated penalties.

Charging requests are linked to the corresponding vehicles and depots. The proposed S2Z data exchange assigns vehicle availability information to the VRP Engine and battery SoC requirements to the Vehicle Consumption Engine.

Optional charging curves describe charging power as a function of battery SoC. The interface also accommodates battery degradation parameters, including charging-rate penalties and calendar ageing.

The depot configuration additionally contains optional temperature-related inputs, represented by an empty array in the example.

Table 15:Charging Scheduling Engine outputs

Output category	**Reported information**
Optimisation results	Solution status, overall feasibility, applied objectives, solver termination reason, relative optimality gap and model size (variables and constraints).
Execution diagnostics	Time required for input validation, model construction, optimisation and output serialisation (ms).
Charging assignment	Charging request and vehicle identifiers, assigned depot and charger, request status and reason codes for unserved or infeasible requests.
Charging schedule	Vehicle arrival and departure times, charger occupancy start and end times, charging start and end times, charging duration and non-charging dead time (min).

Energy and battery state	Charging power (kW), grid energy supplied (kWh) and achieved departure SoC (%).
Depot power utilisation	Configured depot power limit, peak scheduled grid demand and remaining power headroom (kW).
Charging costs	Total charging cost, electricity cost, battery degradation cost where available, and vehicle-level cost allocation (EUR).
Visualisation and diagnostics	Charging and occupancy intervals, vehicle and charger labels, arrival/departure markers, warnings and error messages.

The optimisation status distinguishes between optimal, feasible, partially feasible, infeasible, unavailable and invalid results. Each charging request is also assigned a status indicating whether charging was scheduled, unnecessary, unserved or infeasible. Reason codes provide further information when a request cannot be accommodated.

The charging schedule distinguishes the time during which a charger is occupied from the period of actual energy transfer. This allows non-charging waiting time, charging duration and charger availability to be examined separately.

The energy and cost outputs provide the information required to evaluate alternative charging schedules against the selected objectives. Depot-level results report peak power demand and remaining capacity, while vehicle-level results identify the energy and charging costs associated with each assignment.

The output also includes warnings concerning unsupported or inactive modelling options. In the provided example, the absence of active charging curves is explicitly reported, since the constant-power assumption may underestimate the time required to complete charging.

C.4 Vehicle Consumption Engine

The Vehicle Consumption Engine estimates the mechanical and electrical energy required to complete a vehicle mission and calculates the corresponding battery-state evolution. The model combines route-segment information with vehicle characteristics, payload operations, environmental conditions and the initial battery state.

The available interface is organised around a vehicle simulation object (VehicleSimEngine). Vehicle characteristics are supplied through an instance of the Vehicle class, which contains the physical, powertrain, battery, cargo-space and thermal parameters used by the model.

The segment-level calculation is executed through simulate_segment(segment, vehicle, weather, prev_soc_percent). Route segments and vehicle characteristics can be provided through the corresponding objects or structured JSON representations.

Table 16:Vehicle Consumption Engine input requirements

Input category	Required information
Route segment	Segment identifier (segment_id), distance (distance_km), speed limit (speed_limit_kmh), road gradient (grade_percent) and stop duration (stop_duration_min).
Loading and unloading operations	Cargo manifest, where available, containing the manifest identifier, stop location, parcel information and loading or unloading actions.

Vehicle dynamics	Curb mass, maximum payload, frontal area, drag coefficient, rolling-resistance parameters, maximum speed and acceleration/deceleration limits.
Powertrain	Motor, transmission and inverter efficiencies, regenerative braking efficiency and maximum regeneration power.
Battery	Battery capacity, permitted and initial SoC, nominal voltage, internal resistance, thermal characteristics and heating/cooling limits.
Cargo space and thermal systems	Cargo volume, insulation characteristics, temperature setpoints, eutectic thermal storage capacity, cabin properties and HVAC efficiency.
Environmental conditions	Ambient temperature, wind speed, wind direction and solar irradiance.
Initial battery-state override	Optional prev_soc_percent value used to specify the starting SoC for the segment calculation.

The main fields and their units are described below.

Route segments and cargo operations

Each route segment contains the distance travelled, applicable speed limit, road gradient and any stationary period at the end of the segment. The corresponding fields are distance_km, speed_limit_kmh, grade_percent and stop_duration_min, while segment_id provides a unique reference.

Cargo operations are represented through an optional CargoManifest. Each manifest contains an identifier (manifest_id), location and a list of cargo items. Individual items include item_id, name, mass_kg, volume_m3, temp_requirement_c and action, with the latter indicating whether an item is loaded (LOAD) or unloaded (UNLOAD).

The manifest is used to update vehicle payload at the corresponding stop. This allows the model to account for changes in vehicle mass during the mission rather than assuming a constant payload.

Vehicle and powertrain characteristics

The physical vehicle parameters define the mass and road-load characteristics used in the segment-level energy calculation.

Table 17:Vehicle Physical and Powertrain Input Parameters

Parameter group	Input fields
Mass and dimensions	curb_mass_kg, max_payload_kg, frontal_area_m2
Aerodynamic resistance	drag_coefficient
Rolling resistance	rolling_resistance_base, rolling_resistance_speed_coef
Motion limits	max_speed_kmh, max_accel_m_s2, max_decel_m_s2
Powertrain efficiency	motor_efficiency, transmission_efficiency, inverter_efficiency
Regenerative braking	regen_efficiency, max_regen_kw
Base auxiliary demand	base_aux_kw

The road-load parameters describe the vehicle's rolling and aerodynamic resistance. Speed, acceleration and deceleration limits define the motion constraints used during segment simulation. Motor, transmission and inverter efficiencies determine the conversion between mechanical and electrical energy, while the regeneration parameters limit the energy recovered during braking.

The vehicle object also contains the battery, cargo-bay and HVAC configurations.

Battery and thermal-system characteristics

Battery parameters are provided through the Battery object. They include energy capacity (capacity_kwh), permitted SoC limits (max_soc, min_soc), current SoC (current_soc), nominal voltage (nominal_voltage), internal resistance (internal_resistance_ohm) and battery thermal mass (thermal_mass_j_k). Cooling and heating power are represented through cooling_power_kw and heating_power_kw.

The CargoBay object describes cargo-space volume, insulation heat-transfer characteristics (U·A), temperature setpoints and eutectic plate thermal capacities. The HVACSystem object represents cabin volume, insulation properties, temperature targets and the coefficients of performance used for heating and cooling.

These inputs allow traction energy, battery thermal demand, cabin HVAC and cargo refrigeration to be represented separately within the overall energy calculation.

Environmental conditions and initial battery state

Environmental conditions are supplied through the WeatherData structure. The relevant fields are ambient temperature (ambient_temp_c), wind speed (wind_speed_kmh), wind direction (wind_direction_deg) and solar irradiance (solar_irradiance_w_m2).

The optional prev_soc_percent parameter allows the initial battery SoC to be specified directly for a segment. When provided, it overrides the starting SoC otherwise taken from the vehicle's battery configuration. The permitted numerical range is 0–100%.

C.4.1 Engine outputs

The Vehicle Consumption Engine reports results for each simulated segment. These include the reconstructed motion, payload changes, mechanical energy demand, electrical consumption and battery-state evolution.

Table 18:Vehicle Consumption Engine outputs

Output category	Reported information
Motion and timing	Segment identifier, distance, active driving duration, stop duration and average speed.
Vehicle mass and payload	Gross vehicle mass during travel, payload before the segment stop and payload after loading/unloading operations.
Mechanical energy	Energy associated with acceleration, cruising and braking, together with net wheel-level mechanical energy.
Electrical consumption	Traction energy, regenerative energy recovery, base auxiliaries, cabin HVAC, cargo refrigeration and battery thermal management.
Energy performance	Net battery energy consumption and energy consumption per kilometre.

Battery state and thermal response	Initial and final SoC, battery temperature and resistive heat generation.

The individual output fields are retained in the segment-level results to allow the energy calculation to be examined in greater detail.

Motion and payload results

Motion outputs comprise segment_id, distance_km, duration_min, stop_duration_min and avg_speed_kmh. The reported driving duration includes acceleration, cruising and deceleration, while stationary time is reported separately.

Payload outputs comprise total_mass_kg, payload_kg_before and payload_kg_after. The total mass represents the curb mass and payload during travel. Payload before and after the stop is retained so that loading and unloading operations can be reflected in the following segment.

Mechanical and electrical energy results

Mechanical energy is reported through accel_mech_kwh, cruise_mech_kwh, decel_mech_kwh and total_mech_kwh. These fields distinguish the wheel-level energy required during acceleration and cruising from the mechanical energy available during braking.

Electrical consumption is reported separately for the main vehicle systems:

Table 19:Vehicle Consumption Engine Electrical Energy Output Components

Energy component	Output field	Description
Traction	traction_kwh	Electrical energy drawn for propulsion
Regenerative braking	regen_kwh	Electrical energy recovered during braking
Base auxiliaries	aux_kwh	Vehicle electronics and other base electrical loads
Cabin HVAC	hvac_kwh	Cabin heating and cooling demand
Cargo refrigeration	refrig_kwh	Refrigeration and compressor energy demand
Battery thermal management	tms_kwh	Battery heating and cooling energy demand
Net energy	net_kwh	Total battery energy consumed after accounting for regeneration
Specific consumption	kwh_per_km	Net energy consumption per kilometre

Net battery energy consumption is calculated by combining traction and auxiliary demand and subtracting the recovered regenerative energy:

net_kwh = traction_kwh + aux_kwh + hvac_kwh + refrig_kwh + tms_kwh − regen_kwh

The specific consumption indicator, kwh_per_km, relates the resulting net energy to the distance travelled.

The separation of these components allows propulsion and auxiliary energy requirements to be examined independently, which is relevant for missions involving temperature-controlled cargo or significant cabin and battery thermal demand.

Battery-state and thermal results

The battery outputs include the initial and final State of Charge (soc_before, soc_after), battery temperature (battery_temp_c) and resistive heat generation (battery_heat_gen_kw).

The heat-generation output represents the electrical losses associated with battery internal resistance, expressed through the (I^2R) relationship. The resulting thermal and battery-state information is retained alongside the energy calculation for each segment.

The segment-level outputs provide the detailed consumption and battery information required for mission-level energy assessment. They also support the examination of how route characteristics, payload changes, driving conditions and auxiliary loads contribute to the energy demand of different S2Z vehicle configurations.

Appendix D - Mathematical Formulations and Variable Glossary

Vehicle Routing Problem Engine

The Vehicle Routing Problem (VRP) Engine determines the assignment of customer deliveries to the available vehicle fleet and the sequence in which customers are served. Each vehicle route starts and ends at the depot, with travel distances and times derived from the road network rather than straight-line distances. The formulation combines mandatory customer service with vehicle capacity, fleet availability and route duration constraints.

The routing component uses a lexicographic objective that first minimises the number of active vehicles and subsequently the total distance travelled. Detailed three-dimensional loading feasibility is evaluated by the Cargo Loading Engine rather than embedded directly within the routing optimisation.

Notation

Sets and indices	
C	Set of customers; c and d denote customer indices.
$N = \{0\} \cup C$	Set of routing nodes, where node 0 denotes the depot; u and v denote node indices.
K	Set of available vehicles; k denotes a vehicle index.
E	Set of admissible directed road-network arcs (u,v).
Parameters	
d_{uv}	Real-road distance associated with arc (u,v).
t_{uv}	Travel time associated with arc (u,v).
s_c	Service duration at customer c.
q_c	Aggregate parcel weight required by customer c.
v_c	Aggregate parcel volume required by customer c.
Q_k	Payload capacity of vehicle k.
V_k	Available cargo-volume capacity of vehicle k.
$\bar{T}_k$	Maximum allowed route duration of vehicle k.
M_r	Sufficiently large constant used in the route-order precedence constraint.
Decision variables	
$x_{uvk} \in \{0,1\}$	Equals 1 if vehicle k travels directly from node u to node v; 0 otherwise.
$y_{ck} \in \{0,1\}$	Equals 1 if customer c is assigned to vehicle k; 0 otherwise.
$u_k \in \{0,1\}$	Equals 1 if vehicle k is activated; 0 otherwise.
$r_{ck} \in \mathbb{Z}_{\geq 0}$	Route-order variable giving the position of customer c in the route of vehicle k.

Customer service is mandatory. The optimisation is therefore lexicographic: the number of active vehicles is minimised first and, among solutions using the same fleet size, the total real-road distance is minimised second.

$$\min_{\text{lex}} (Z_{\text{fleet}},\ Z_{\text{road}}) \tag{1}$$

$$Z_{\text{fleet}} = \sum_{k \in K} u_k \tag{2}$$

$$Z_{\text{road}} = \sum_{k \in K} \sum_{(u,v) \in E} d_{uv}\, x_{uvk} \tag{3}$$

Every customer must be assigned to exactly one vehicle, and a customer can only be assigned to a vehicle that is active:

$$\sum_{k \in K} y_{ck} = 1 \qquad \forall c \in C \tag{4}$$

$$y_{ck} \leq u_k \qquad \forall c \in C,\ k \in K \tag{5}$$

For each active vehicle, the route must depart from and return to the depot. Customer-level flow conservation then links assignment and routing decisions:

$$\sum_{v \in C} x_{0vk} = u_k, \qquad \sum_{u \in C} x_{u0k} = u_k \qquad \forall k \in K \tag{6}$$

$$\sum_{\substack{v \in N \\ v \neq c}} x_{cvk} = y_{ck}, \qquad \sum_{\substack{u \in N \\ u \neq c}} x_{uck} = y_{ck} \qquad \forall c \in C,\ k \in K \tag{7}$$

The aggregate weight and volume assigned to each vehicle must remain within the corresponding vehicle capacities:

$$\sum_{c \in C} q_c\, y_{ck} \leq Q_k u_k \qquad \forall k \in K \tag{8}$$

$$\sum_{c \in C} v_c\, y_{ck} \leq V_k u_k \qquad \forall k \in K \tag{9}$$

The aggregate volume constraint is a necessary routing-level capacity check. It does not guarantee that the assigned parcels can be physically arranged inside the cargo bay. Detailed three-dimensional feasibility is evaluated separately by the Cargo Loading Engine.

The total route duration, including road travel and customer service, must remain within the available operating horizon:

$$\sum_{(u,v) \in E} t_{uv}\, x_{uvk} + \sum_{c \in C} s_c\, y_{ck} \leq \bar{T}_k u_k \qquad \forall k \in K \tag{10}$$

Finally, route-order variables are used to exclude disconnected subtours and maintain a valid customer sequence. The route-order variable represents the position of a customer in a vehicle route, while a valid big-M constant activates the precedence relation only when the corresponding routing arc is selected:

$$r_{dk} \geq r_{ck} + 1 - M_r(1 - x_{cdk}) \qquad \forall c \neq d,\ c,d \in C,\ k \in K \tag{11}$$

Equations (1)-(11) describe the routing-level decisions used to determine vehicle activation, customer assignment and visit sequence. Detailed three-dimensional placement, structural support, axle-load distribution and unloading accessibility are intentionally left to the Cargo Loading Engine, which can feed loading infeasibilities back to the routing process.

Cargo Loading Engine

The Cargo Loading Engine evaluates whether the parcels assigned to each vehicle can be accommodated within its cargo compartment while satisfying geometric, structural, operational and vehicle weight restrictions. Unlike the routing-level capacity checks,

which consider aggregate parcel weight and volume, the loading model determines the actual three-dimensional position and orientation of each parcel.

The formulation is based on a finite set of admissible placement candidates. Each candidate combines a permitted parcel orientation with a position inside the cargo compartment. When the engine operates after the VRP, vehicle assignments and customer visit sequences are treated as fixed inputs.

Notation

Sets and indices	
C	Set of customers, indexed by c and d.
K	Set of available vehicles, indexed by k.
I	Set of parcels, indexed by i, j and l.
I_c	Set of parcels delivered to customer c.
$c(i)$	Customer associated with parcel i.
O_i	Set of admissible orthogonal orientations of parcel i.
A_{ik}	Set of feasible placement candidates for parcel i in vehicle k.
Ω_{kov}	Set of incompatible placement-candidate pairs with positive-volume overlap.
S_k	Set of admissible direct-support candidate pairs.
D_k	Set of direct geometric blocker relations used for unloading accessibility.
$G = \{x_-, x_+, y_-, y_+\}$	Set of support-edge directions used by the overhang guard.
Parameters	
q_i	Weight of parcel i.
$L_k,\ W_k,\ H_k$	Cargo-bay length, width and height of vehicle k.
Q_k	Maximum payload of vehicle k.
W_{k0}	Empty-vehicle mass represented by the measured empty axle loads.
$F_{k0},\ R_{k0}$	Empty-vehicle front and rear axle loads.
$\bar{F}_k,\ \bar{R}_k$	Maximum front and rear axle loads of vehicle k.
$\sigma_i \in \{0,1\}$	Equals 1 if parcel i may support another parcel; 0 otherwise.
α	Minimum supported fraction of the upper parcel base area; the supplied MILP uses 0.75.
$\bar{H}$	Maximum number of distinct parcels that may be displaced during one customer unloading.
$(\bar{x}_{ika}, \bar{y}_{ika}, \bar{z}_{ika})$	Coordinates of placement candidate a in A_ik.
$(\ell_{ika}, w_{ika}, h_{ika})$	Oriented parcel dimensions associated with candidate a.
$A_{ika\mathrm{base}}$	Base area of the upper parcel candidate.
$A_{ika,jkb\mathrm{sup}}$	Precomputed horizontal contact area for an admissible support pair.
$f_{ika} \in \{0,1\}$	Equals 1 if candidate (i,k,a) lies on the vehicle floor.
$\chi_{ika,jkb}$	Equals 1 if the lower candidate supports the upper candidate base-centre projection.
$\chi_{ika,jkbg}$	Equals 1 if the support contact satisfies edge condition g in G.
ρ_{ikaR}	Rear-axle reaction contributed by parcel i at candidate a.
Fixed VRP inputs	
u_k	Vehicle-activation indicator fixed by the VRP solution.
y_{ck}	Customer-to-vehicle assignment indicator fixed by the VRP solution.
r_{ck}	Route-order value for customer c on vehicle k, fixed by the VRP solution.

e_{cdk}	Precedence indicator: customer c is visited before customer d by vehicle k.
Decision variables	
$p_{ika} \in \{0,1\}$	Equals 1 if placement candidate a is selected for parcel i in vehicle k.
$z_{ika,jkb} \in \{0,1\}$	Equals 1 if selected lower candidate (j,k,b) directly supports selected upper candidate (i,k,a).
$h_{jck\text{move}} \in \{0,1\}$	Equals 1 if parcel j must be displaced to complete unloading of customer c from vehicle k.
$m_{ck\text{disp}} \in \mathbb{Z}_{\geq 0}$	Number of distinct parcels displaced during unloading of customer c from vehicle k.
$w_{ikac\text{rem}} \in \{0,1\}$	Equals 1 if candidate (i,k,a) remains after customer c has been unloaded.
P_k, F_k, R_k	Departure cargo mass and total front/rear axle loads.
$P_{ck\text{rem}}$, $F_{ck\text{rem}}$, $R_{ck\text{rem}}$	Remaining cargo mass and axle loads after unloading customer c.
δ_{kx}, δ_{ky}	Complete-vehicle CoG deviation variables relative to the empty-vehicle reference.
$L_{k\text{env}}$, $W_{k\text{env}}$, $H_{k\text{env}}$	Dimensions of the occupied cargo envelope.

For a fixed routing solution, loading is primarily a feasibility problem. When several feasible layouts exist, the lower-level criteria of the integrated MILP are retained lexicographically: unloading effort is minimised first, followed by complete-vehicle centre-of-gravity deviation and occupied-envelope compactness.

$$\min_{\text{lex}}\left(Z_{\text{move}}, Z_{\text{CoG}}, Z_{\text{compact}}\right) \tag{12}$$

The corresponding lower-level objective components are defined as follows:

$$\begin{aligned} Z_{\text{move}} &= \sum_{k\in K}\sum_{c\in C}\sum_{j\in I\setminus I_c} h_{jck\text{move}} \\ Z_{\text{CoG}} &= \sum_{k\in K}\left(\frac{\delta_{kx}}{(W_{k0}+Q_k)L_k} + \frac{\delta_{ky}}{(W_{k0}+Q_k)W_k}\right) \\ Z_{\text{compact}} &= \sum_{k\in K}\left(\frac{L_{k\text{env}}}{L_k} + \frac{W_{k\text{env}}}{W_k} + \frac{H_{k\text{env}}}{H_k}\right) \end{aligned} \tag{13}$$

For each parcel-vehicle pair, the candidate domain contains only placements whose oriented dimensions remain inside the cargo-bay boundaries:

$$\begin{aligned} 0 \leq \bar{x}_{ika}, &\qquad \bar{x}_{ika} + \ell_{ika} \leq L_k \\ 0 \leq \bar{y}_{ika}, &\qquad \bar{y}_{ika} + w_{ika} \leq W_k \\ 0 \leq \bar{z}_{ika}, &\qquad \bar{z}_{ika} + h_{ika} \leq H_k \end{aligned} \tag{14}$$

Exactly one feasible placement is selected for every parcel assigned to a vehicle:

$$\sum_{a\in A_{ik}} p_{ika} = y_{c(i)k} \qquad \forall i \in I,\ k \in K \tag{15}$$

Positive-volume overlap between two selected candidates is prohibited through the precomputed incompatibility set:

$$p_{ika} + p_{jkb} \leq 1 \qquad \forall\big((i,k,a),(j,k,b)\big) \in \Omega_{\text{kov}},\ i<j \tag{16}$$

A support relation can be active only when both placements are selected and the lower parcel is stackable:

$$z_{ika,jkb} \le p_{ika}$$
$$z_{ika,jkb} \le p_{jkb} \quad (17)$$
$$z_{ika,jkb} \le \sigma_j$$

For every elevated selected parcel, the selected supporters must provide the minimum supported base-area fraction:

$$\sum_{(j,b):((i,k,a),(j,k,b))\in S_k} A_{ika,jkb\mathrm{sup}}\, z_{ika,jkb} \ge \alpha A_{ika\mathrm{base}}(1-f_{ika})p_{ika} \quad (18)$$

Centre support and one-sided overhang are enforced through precomputed support indicators:

$$\sum_{(j,b):((i,k,a),(j,k,b))\in S_k} \chi_{ika,jkb}\, z_{ika,jkb} \ge (1-f_{ika})p_{ika}$$
$$\sum_{(j,b):((i,k,a),(j,k,b))\in S_k} \chi_{ika,jkbg}\, z_{ika,jkb} \ge (1-f_{ika})p_{ika} \qquad \forall g \in G \quad (19)$$

Support chronology preserves residual stability. If parcel j supports parcel i, the supporter cannot be removed before the supported parcel:

$$r_{c(i)k} \le r_{c(j)k} + |C|\left(1 - z_{ika,jkb}\right) \qquad \forall\left((i,k,a),(j,k,b)\right) \in S_k \quad (20)$$

Unloading accessibility is represented by direct blockers and their transitive closure. For the corresponding precomputed blocker relations, the move variables are activated as follows:

$$h_{jck\mathrm{move}} \ge p_{ika} + p_{jkb} + e_{c,c(j),k} - 2$$
$$h_{jck\mathrm{move}} \ge h_{lck\mathrm{move}} + p_{lka} + p_{jkb} + e_{c,c(j),k} - 3 \quad (21)$$

The number of distinct displaced parcels is then limited for each complete customer unloading batch:

$$m_{ck\mathrm{disp}} = \sum_{j\in I\setminus I_c} h_{jck\mathrm{move}}$$
$$m_{ck\mathrm{disp}} \le \bar{H}\, y_{ck} \qquad \forall c \in C,\ k \in K \quad (22)$$

Departure axle reactions are obtained from the selected parcel placements and constrained by the admissible axle limits:

$$P_k = \sum_{i\in I}\sum_{a\in A_{ik}} q_i\, p_{ika}$$
$$R_k = R_{k0}u_k + \sum_{i\in I}\sum_{a\in A_{ik}} \rho_{ikaR}\, p_{ika}$$
$$F_k = F_{k0}u_k + P_k - \sum_{i\in I}\sum_{a\in A_{ik}} \rho_{ikaR}\, p_{ika} \quad (23)$$
$$0 \le F_k \le \bar{F}_k u_k, \qquad 0 \le R_k \le \bar{R}_k u_k$$

Axle feasibility is also checked after every realised customer unloading state:

$$
\begin{aligned}
P_{ck\mathrm{rem}} &= \sum_{i \notin I_c} \sum_{a \in A_{ik}} q_i \, w_{ikac\mathrm{rem}} \\
R_{ck\mathrm{rem}} &= R_{k0} y_{ck} + \sum_{i \notin I_c} \sum_{a \in A_{ik}} \rho_{ikaR} \, w_{ikac\mathrm{rem}} \\
F_{ck\mathrm{rem}} &= F_{k0} y_{ck} + P_{ck\mathrm{rem}} - \sum_{i \notin I_c} \sum_{a \in A_{ik}} \rho_{ikaR} \, w_{ikac\mathrm{rem}} \\
0 \le F_{ck\mathrm{rem}} &\le \bar{F}_k y_{ck}, \qquad 0 \le R_{ck\mathrm{rem}} \le \bar{R}_k y_{ck}
\end{aligned}
\tag{24}
$$

Charging Scheduling Engine

The Charging Scheduling Engine allocates vehicles to compatible chargers and determines the timing of charging sessions within the depot planning horizon. Its objective is to satisfy vehicle departure energy requirements while respecting charger availability, charging windows, handling times and depot electrical capacity. Electricity tariffs and optional battery degradation costs are also considered when determining the preferred charging schedule.

The mathematical formulation is based on feasible charging modes generated before optimisation. Each mode combines a compatible charger class with an admissible charging-start interval and the corresponding charging duration and power profile. The resulting mixed-integer model selects charging modes, schedules sessions and coordinates the use of shared charging infrastructure.

Notation

Sets and indices	
R	Set of schedulable charging requests, indexed by r.
C	Set of charger classes, indexed by c.
M_r	Set of feasible charging modes for request r. A mode represents one compatible charger class and one admissible charging-start interval.
L_m	Set of constant-power charging segments associated with mode m, indexed by ℓ.
A	Set of admissible predecessor-successor request pairs that can be sequenced on a common charger class.
B	Finite set of relevant segment-start events used for exact depot-power evaluation.
Parameters	
C_{ruse}	Usable battery capacity of the vehicle associated with request r (kWh).
SOC_{rarr}	State of charge of request r at arrival (%).
SOC_{rreq}	Required state of charge of request r at departure (%).
a_r	Arrival time of request r, measured from the start of the planning horizon (min).
d_r	Required departure time of request r, measured from the start of the planning horizon (min).
g_{in}	Plug-in handling time (min).
g_{out}	Plug-out handling time (min).
$c(m)$	Charger class selected by mode m.
n_c	Number of physical charger units available in charger class c.

l_m	Earliest feasible charging-start time of mode m (min).
u_m	Latest feasible charging-start time of mode m (min).
τ_m	Complete charging duration of mode m (min).
$p_{m\ell}$	Grid power of charging segment ℓ of mode m (kW).
$P_{\max}$	Depot grid-power limit (kW).
$K_m(s)$	Electricity cost of mode m when charging starts at time s, obtained from the applicable tariff intervals.
$u_{rm\ell b}$	Binary event-activity indicator equal to 1 when charging segment ℓ of selected mode m for request r is active at event b.
κ_{rm}	Exact piecewise-linear electricity-cost contribution of mode m for request r; zero when that mode is not selected.
π_{grid}	Penalty per kW applied to soft exceedance of the nominal depot power limit.
D_m	Battery-degradation cost associated with selecting mode m, when the required degradation data are available.
M	Sufficiently large constant used in the charger-sequencing linearisation.
Decision variables	
x_{rm}	Binary variable equal to 1 if request r is assigned to mode m ∈ M_r.
z_r	Binary served indicator for request r.
s_{rm}	Charging-start contribution of mode m for request r; zero when the mode is not selected.
s_r	Charging start time of request r (min).
δ_r	Dead time between the end of plug-out and vehicle departure (min).
$y_{rr'}$	Binary variable equal to 1 if request r′ directly follows request r on the same charger chain.
f_{rc}	Binary variable equal to 1 if request r is the first session on a physical charger chain of class c.
e	Permitted exceedance above the nominal depot power limit when that constraint is configured as soft (kW).
P_{peak}	Depot peak charging power when peak-power minimisation is active (kW).

The battery energy that must be added for each charging request is determined directly from the usable battery capacity and the required increase in state of charge:

$$E_{\text{rreq}} = C_{\text{ruse}} \frac{\text{SOC}_{\text{rreq}} - \text{SOC}_{\text{rarr}}}{100} \tag{25}$$

Each request may select at most one complete uninterrupted charging mode. The served indicator is therefore the sum of the selected mode variables:

$$z_r = \sum_{m \in M_r} x_{rm} \leq 1, \qquad \forall r \in R \tag{26}$$

The request charging-start time is linked to the mode-specific start contributions, and each selected contribution must remain inside the preprocessed feasible interval of that mode:

$$s_r = \sum_{m \in M_r} s_{rm}, \qquad \forall r \in R \tag{27}$$

$$l_m x_{rm} \le s_{rm} \le u_m x_{rm}, \qquad \forall r \in R,\ m \in M_r \tag{28}$$

The charging duration of a served request is determined by the selected mode. Dead time is constrained to the non-negative interval between the end of plug-out and the required departure time:

$$\delta_r \ge d_r z_r - g_{\text{out}} z_r - s_r - \sum_{m \in M_r} \tau_m x_{rm}, \qquad \forall r \in R \tag{29}$$

$$\delta_r \ge 0, \qquad \forall r \in R \tag{30}$$

Physical charger availability is represented through non-overlapping session chains. For every charger class, the number of chain heads cannot exceed the number of available physical units, and each served request must either start one chain or have exactly one predecessor:

$$\sum_{r \in R} f_{rc} \le n_c, \qquad \forall c \in C \tag{31}$$

$$\sum_{c \in C} f_{rc} + \sum_{q:(q,r) \in A} y_{qr} = z_r, \qquad \forall r \in R \tag{32}$$

$$\sum_{q:(r,q) \in A} y_{rq} \le z_r, \qquad \forall r \in R \tag{33}$$

When request r′ follows request r on the same charger chain, the start of r′ must occur after charging and plug handling for r have been completed. The corresponding big-M sequencing condition is:

$$s_r + \sum_{m \in M_r} \tau_m x_{rm} + (g_{\text{in}} + g_{\text{out}}) z_r - s_q \le M\big(1 - y_{rq}\big), \qquad \forall (r,q) \in A \tag{34}$$

Depot power is evaluated using the actual constant-power segments of the selected charging curves. At each relevant event, the simultaneous segment-level grid power must remain below the depot limit, with optional exceedance e when the constraint is configured as soft:

$$\sum_{r \in R} \sum_{m \in M_r} \sum_{\ell \in L_m} p_{m\ell}\, u_{rm\ell b} \le P_{\max} + e, \qquad \forall b \in B \tag{35}$$

$$\sum_{r \in R} \sum_{m \in M_r} \sum_{\ell \in L_m} p_{m\ell}\, u_{rm\ell b} \le P_{\text{peak}}, \qquad \forall b \in B \tag{36}$$

The optimisation is lexicographic. The first level maximises the number of served requests; subsequent levels minimise the objectives listed in objective.priority in the order supplied by the user:

$$\max \sum_{r \in R} z_r \tag{37}$$

$$J_{\text{dead}} = \sum_{r \in R} \delta_r \tag{38}$$

$$J_{\text{energy}} = \sum_{r \in R} \sum_{m \in M_r} \kappa_{rm} + \pi_{\text{grid}} e \tag{39}$$

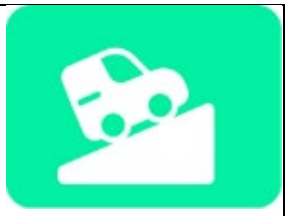
$$J_{\text{deg}} = \sum_{r \in R} \sum_{m \in M_r} D_m \, x_{rm} \tag{40}$$

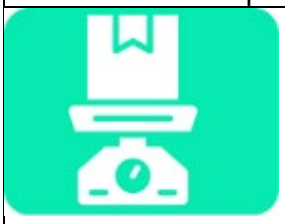
$$J_{\text{peak}} = P_{\text{peak}} \tag{41}$$

Vehicle Consumption ModelEngine

The engine models an eLCV's real-world energy use by combining five core factors:

Elevation profile (road gradient): Uphill grades sharply increase consumption; downhill segments allow regenerative recovery. For example, one study found an 8% uphill requires ~81 Wh whereas an 8% downhill yields only ~8 Wh net use . (Positive slopes add gravitational load; negative slopes engage regenerative braking). Gravitational effects dominate, steep climbs significantly increase consumption, while coasting downhill can recuperate a fraction back into the battery.





Vehicle payload (mass): Heavier cargo means more rolling resistance and inertia. Added mass directly raises energy use, especially during acceleration and grades. Physics models calculate rolling resistance as proportional to vehicle and cargo weight, and empirical analysis confirms that weight is a key driver. When total mass increases, consumption rises in proportion

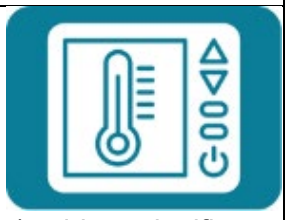
Driving speed profiles: Higher speeds significantly increase aerodynamic drag (~speed^2) and rolling losses. At low speeds (<30 kph), consumption is modest, but above ~30 kph, drag dominates and energy use increases steeply. Thus, steady highway driving typically uses more battery per km than lower-speed suburban runs. (Note: unlike ICE vehicles, EVs partially offset stop-and-go via regen, so net city vs highway effects differ – see next point.) The model uses the planned speed profile (urban vs interurban) to weight drag and brake losses per segment.





Stop frequency and acceleration: Frequent stops require repeated acceleration, which draws high power. Each stop-start cycle imposes kinetic energy losses (some recovered by regeneration). Notably, EVs *regain* energy when braking or coasting (via regenerative braking), making city driving relatively more efficient than a comparable ICE vehicle. However, every acceleration from standstill still consumes extra kWh; the engine counts each stop (from the delivery/pickup schedule and traffic lights) and models the work to accelerate back to speed, minus the recovered power. Testing shows that rapid acceleration and deceleration significantly impact consumption, so the engine adjusts according to the planned stop-layout.

Ambient temperature (HVAC loads): Heating or cooling can account for a large share of trip energy, especially in extreme weather. For example, cold climates can **reduce EV range by up to ~40%** (in one test, range fell 41% at -7° C) due to heater draw and battery chemistry losses. The model uses weather forecasts or historical temperature values to estimate HVAC power. Heating in winter (or air conditioning in very hot weather) adds a significant kWh load. DOE studies highlight that EV HVAC demand grows sharply as outside temperature deviates from 20–25° C comfort levels. In practice, we model HVAC as a temperature-dependent auxiliary load on top of propulsion energy.



These factors are combined in a physics-based simulation. The route is broken down into segments (or timesteps) with the distance, elevation change, and speed/stop pattern known in the equations. For each step, the model calculates the vehicle's change in kinetic and potential energy (speed and altitude) and sums resistive losses: aerodynamic drag, rolling resistance, auxiliary loads (HVAC, lights, etc.). Regenerative braking from deceleration or downhill is multiplied by an efficiency factor and subtracted from the losses. Formally, the vehicle's energy status is updated at each step by adding energy gain (if any) and subtracting energy used. In other words, whenever the LCV accelerates or climbs, it *consumes* battery energy; when it coasts or descends, it *recovers* some. The net result of all segments is the **total kWh consumed** on the route and a detailed SoC profile. At completion, the engine reports the cumulative kWh and the battery's state of charge at each point.

Longitudinal Vehicle Dynamics

Mechanical power at the wheels is derived from Newton's second law for a vehicle of total mass m (including cargo) at speed v on a road of slope angle θ (positive uphill). Total resistance force is the sum of rolling resistance, aerodynamic drag, grade (gravitational) force, and inertial force:

$$F_{\text{res}} = F_{\text{roll}} + F_{\text{drag}} + F_{\text{grade}} + F_{\text{inertia}}$$

Instantaneous mechanical power at the wheels is $P_{wheel} = F_{res} \cdot v\,(positive = traction, negative = braking)$. Segment mechanical energy is obtained by integrating power over time, or, under a quasi-steady assumption, by summing contributions from the acceleration, cruise and deceleration phases.

Rolling Resistance, Aerodynamic Drag and Road Grade

To accurately resolve the instantaneous resistance force F_{res}, the total load is decomposed into its physical constituents: rolling resistance, aerodynamic drag, gravitational force due to road grade, and inertial resistance.

Rolling resistance at the tire-road interface is captured using a two-term model that accounts for both static deformation and speed-dependent hysteresis losses:

$$F_{\text{roll}} = mg(c_0 + c_1 v)\cos\theta \qquad (42)$$

where

c_0 is the static rolling resistance coefficient (dimensionless),

and c_1 is the velocity-dependent rolling resistance coefficient (s/m)

This rolling friction acts alongside aerodynamic drag, which opposes vehicle motion quadratically relative to speed. It is modeled using the vehicle's frontal area A and drag coefficient C_d:

$$F_{\text{drag}} = \frac{1}{2}\rho_{\text{air}} C_d A v^2 \qquad (43)$$

where ρ represents ambient air density, taken as $\rho_{\text{air}} = 1.225\ \text{kg/m}^3$ at 15 °C.

In addition to friction and drag, the vehicle experiences a gravitational force component whenever traversing inclined terrain:

$$F_{\text{grade}} = mg\sin\theta \tag{44}$$

Finally, during dynamic changes in speed, acceleration is resisted by both the linear mass of the vehicle and the rotational inertia of rotating components (wheels, motor and drivetrain). This inertial force is expressed as:

$$F_{\text{inertia}} = m_{\text{eff}}\, a \tag{45}$$

where $m_{\text{eff}} = m \cdot \lambda$ accounts for rotational inertia of wheels and drivetrain ($\lambda \approx 1.04$).

The instantaneous mechanical power at the wheels is $P_{\text{wheel}} = F_{\text{res}}\, v$ (positive = traction, negative = braking).

Acceleration, Deceleration and Stop-and-Go Operation

Each segment is simulated through discrete motion phases: acceleration from the previous speed to a feasible cruise speed, cruise at constant speed, deceleration to zero if the segment ends in a stop, extra stop-go cycles (e.g. traffic lights, modelled as additional accel/decel pairs), and dwell (stationary, auxiliary loads only).

The feasible cruise speed is derived kinematically from the target speed and available distance, capped by the segment's target speed. Mechanical energy per phase is

$$E_{\text{mech}} = \left[(F_{\text{roll}} + F_{\text{drag}} + F_{\text{grade}}) \cdot d\right] + \frac{1}{2} m_{\text{eff}} (v_{\text{end}}^2 - v_{\text{start}}^2) \tag{46}$$

with forces evaluated at the average speed of the phase.

Motor and Drivetrain Efficiency

The electric motor is modelled with a static efficiency map η_m, interpolated from a 6×6 table as a function of normalised torque ($\frac{T}{T_{\max}}$) and speed ($\frac{n}{n_{\max}}$); the same map is used for motoring and regeneration, as follows:

$$\eta_m = f\left(\frac{T}{T_{\max}},\ \frac{n}{n_{\max}}\right) \tag{47}$$

where T is motor torque (Nm), n motor speed (RPM), and $T_{\max}, n_{\max}$ are peak values. The same map is used for both motoring and regeneration.

A cold-weather penalty factor $\chi(T_{\text{amb}})$ models increased internal resistance at low temperatures:

$$\chi(T_{\text{amb}}) = \begin{cases} 1.0, & \text{if } T_{\text{amb}} \geq 15°\text{C} \\ 1.0 + 0.010 \cdot (15 - T_{\text{amb}}), & \text{if } 0°\text{C} \leq T_{\text{amb}} < 15°\text{C} \\ 1.15 + 0.015 \cdot |T_{\text{amb}}|, & \text{if } T_{\text{amb}} < 0°\text{C} \end{cases} \tag{48}$$

Together with battery capacity derating (Section 5.1), this reproduces the observed ~40 % range loss at −7 °C.

Transmission and Inverter Losses

The transmission (gearbox and differential) is modeled with a constant mechanical efficiency η_{trans} (transmission_eff, default 0.97).

Inverter losses are now fully implemented in the code base:

- Inverter Efficiency: Modeled as a separate conversion efficiency factor η_{inv} (inverter_eff, default 0.95) applied alongside motor efficiency during motoring and regeneration.
- Inverter Standby Power: Active power consumption during movement and dwell is modeled by inverter_standby_loss_kw (default 0.05 kW).

During traction ($P_{\text{wheel}} \geq 0$), wheel mechanical power is converted to battery terminal electrical power by:

$$P_{\text{batt, traction}} = \left(\frac{P_{\text{wheel}}}{\eta_{\text{motor}} \cdot \eta_{\text{trans}} \cdot \eta_{\text{inv}}} \cdot \chi(T_{\text{amb}}) \right) + P_{\text{inv_standby}} \tag{49}$$

Regenerative Braking

During braking ($P_{\text{wheel}} < 0$), recovered battery power is

$$P_{\text{batt}} = -\min\left(|P_{\text{wheel}}| \cdot \eta_{\text{regen}} \eta_{\text{trans}}, P_{\text{regen,max}}, P_{\text{charge,limit}} \right) \tag{50}$$

where η_{regen} is regenerative braking efficiency and $P_{\text{regen,max}}$ a system-level power cap; the battery's charge power limit is imposed as an additional constraint. Regeneration is cut off once SoC exceeds a configurable threshold (default 0.95) to protect the battery.

Battery Energy and State-of-Charge Modelling

The battery is represented as an equivalent circuit model: open-circuit voltage V_{OC} in series with internal resistance R_{int}, giving terminal voltage V_{term}. V_{OC} varies linearly with SoC between V_{empty} (v_oc_empty_v) and V_{full} (v_oc_full_v):

$$V_{\text{OC}}(\text{SoC}) = V_{\text{empty}} + \text{SoC} \cdot \left(V_{\text{full}} - V_{\text{empty}} \right) \tag{51}$$

Internal resistance depends on cell temperature and State of Health (SoH):

$$R_{\text{int}}(T_{\text{cell}}) = R_{\text{ref}} \cdot f_{\text{SoH, R}} \cdot (1 + \alpha_R \cdot \max(0{,}20 - T_{\text{cell}})) \tag{52}$$

For a terminal power demand P_{batt} (kW), current I (Amperes, positive = discharge) is calculated by solving $P_{\text{batt}} \cdot 1000 = V_{\text{OC}} \cdot I - R_{\text{int}} \cdot I^2$:

$$I = \frac{V_{OC} - \sqrt{\max\left(0, V_{OC}^2 - 4 \cdot R_{int} \cdot P_{batt} \cdot 1000\right)}}{2 \cdot R_{int}} \quad (52)$$

Temperature and degradation effects on usable capacity are governed by:

$$C_{derated} = C_{nominal} \cdot SoH \cdot f_{SoH,\,cap} \cdot f_{temp}(T_{cell}) \quad (53)$$

where $f_{temp}(T_{cell})$ is a piecewise linear derating function (e.g., 1.0 at $20°C$, 0.82 at $0°C$, 0.60 at $-20°C$). SoH capacity degradation is explicitly accounted for via soh and soh_capacity_fade_factor.

SoC updates over duration Δt (hours) using Coulomb/energy integration:

$$SoC_{new} = \text{clamp}\left(SoC_{old} - \frac{P_{actual,\,kW} \cdot \Delta t}{C_{derated}}, 0.0, 1.0\right) \quad (54)$$

The SoC is kept strictly within $[0,1]$ and compared against a user-defined reserve threshold (min_soc_fraction, default 0.10).

Thermal and Auxiliary Energy Demand

Auxiliary electrical loads are added directly to traction demand at the battery level:

$$P_{aux,\,total} = P_{base} + P_{inv_standby} + P_{HVAC} + P_{cargo_refrig} + P_{TMS} \quad (55)$$

They include:

- **Cabin HVAC**: heat-pump model, with the electrical power estimated as:
  - Heating Load: $P_{th} = 0.15 \cdot \Delta T \quad (\text{if } \Delta T > 0)$
  - Cooling Load: $P_{th} = 0.10 \cdot |\Delta T| \quad (\text{if } \Delta T < 0)$

  (56)

  Then the electrical power is:

  $$P_{HVAC} = \frac{P_{therm}}{COP(T_{amb})} \quad (57)$$

  The Coefficient of Performance (COP) degrades linearly below 0 °C, reaching 1.0 at −15 °C (resistive heating).

- **Multi-zone cargo refrigeration**: with the cargo body is divided into up to three zones frozen (18 °C), refrigerated (+4 °C), and dry , each with heat gain $Q_{gain} = U \cdot A \cdot \max(0,\, T_{amb} - T_{set})$.

  Each zone has:
  - Surface area A and overall heat transfer coefficient U $(W/m^2 \cdot K)$
  - Current cargo mass and specific heat
  - Eutectic (phase change material) storage capacity $Q_"eut"$ (kWh)

The refrigeration system uses a single compressor with COP that depends on ambient temperature:

$$COP(T_amb) = COP_0 - \beta \cdot max(0, T_amb - 20\,°C) \quad (58)$$

Cooling priority operates as follows:

1. Thermal inflow $Q_{\text{total}} = \sum \dot{Q}_{\text{zone}} \cdot \Delta t$ is absorbed first by phase-change eutectic plates (zero electrical draw).
2. Any residual heat requires compressor work: $P_{\text{comp}} = \frac{Q_{\text{residual}}}{\Delta t \cdot \text{COP}_{\text{comp}}(T_{\text{amb}})}$.
3. Ventilation fan base load (P_{vent}, default 0.15 kW) is continuously added. Pre-cooling depot energy (sensible cargo mass cooling + eutectic plate charge) is calculated separately via pre_cool_charge_kwh() and drawn from depot mains power, avoiding vehicle battery draw.

Pre cooling at depot is modelled separately: the energy needed to cool the cargo and charge the eutectic plates is drawn from the mains, not the vehicle battery.

- Battery thermal management (TMS): proportional controller keeping cell temperature within 15–35 °C, capped at 3 kW.

$$P_{\text{TMS}} = \begin{cases} 0 & 15 \le T_{\text{cell}} \le 35 \\ 0.05 \cdot \Delta T & \text{otherwise, capped at 3 kW} \end{cases} \quad (59)$$

- Base Auxiliaries and Dwell Losses: Base electronics/lighting load (P_{base}, default 0.5 kW) plus standby losses operate during travel and dwell phases. During dwell stops, traction power is zero, and only auxiliary loads consume battery energy.

Because the dwell phase already applies only auxiliary loads (no traction), stop/dwell energy losses are calculable within the existing model -this appears to resolve the open question of whether dwell energy losses can be computed, though it has not yet been validated against measured stop data.

Payload Evolution

Per-stop cargo operations are represented by a CargoManifest attached to each RouteSegment. It stores explicit entries of (zone_name, load_kg, unload_kg) allowing cross-docking operations and dynamic mass updates across multiple thermal zones in a single stop.

The manifest is applied at the conclusion of a segment's travel and dwell phases via apply_manifest(), updating individual zone masses and overall vehicle curb/total mass for subsequent segments:

$$m_{\text{cargo, zone, new}} = \max\left(0, m_{\text{cargo, zone, old}} - m_{\text{unload}} + m_{\text{load}}\right) \quad (60)$$

Stop-to-Stop Energy Assessment

The engine reports granular metrics at the micro-segment level (SegmentResult) and full-route level (RouteResult). Additionally, stop-to-stop leg provenance tracking is supported via explicit segment fields:

- source_leg_index: Index of the high-level origin-destination leg.
- source_from_stop_id: Identifier for the starting stop.
- source_to_stop_id: Identifier for the destination stop.

These fields allow segment-level mechanical and electrical results to be aggregated directly into stop-to-stop leg totals.

Route Feasibility and Minimum-SoC Assessment

After simulating all route segments, the engine assesses whether SoC drops below the minimum reserve threshold (SoC_{min}) at any point along the route. The engine identifies the exact index of the critical segment where the breach first occurs and computes the exact battery shortfall in kWh:

$$\text{Shortfall (kWh)} = \left(SoC_{min} - SoC_{min_observed}\right) \cdot C_{derated} \quad (61)$$

Assessing feasibility based on the global minimum SoC reached during the route prevents masking shortfalls on routes where regenerative braking raises SoC near the end of a trip.

When a route is marked infeasible, automated remediation suggestions are generated:

- Inserting intermediate charging stops prior to the critical segment.
- Reducing payload using first-order mass sensitivity estimates.
- Pre-conditioning the cabin and fully charging eutectic refrigeration plates while plugged in at the depot.

Real-Time State Estimation and Model Updating

A 1D Kalman Filter (SoCKalmanFilter) fuses physics-predicted SoC updates with live telemetry measurements (SoC and energy consumption), filtering noise and correcting state estimates during operation. The RealTimeEngine uses this updated state to re-simulate remaining segments, enabling dynamically updated routing and opportunistic charging.

In parallel, a data-driven TelemetryCalibrator optimizes key physical parameters ($c_{r,0}$, C_d, η_{regen_mod}) using historical trip telemetry. It utilizes SciPy minimize (BFGS / L-BFGS-B) with a pure-Python gradient descent fallback, adjusting model coefficients to align with driver behavior and vehicle wear over time.