NIGHT-TIME CARGO DELIVERY PROCESS AT KLM CARGO

REDESIGNING THE NIGHT-TIME DELIVERY PROCESS AND PROCEDURE AT CARGO TERMINAL 1

TIL5060: THESIS PROJECT

DYLAN BRAVO

TU Delft

AIRFRANCE KLM Martinair CARGO

NIGHT-TIME CARGO DELIVERY PROCESS AT KLM CARGO

REDESIGNING THE NIGHT-TIME DELIVERY PROCESS AND PROCEDURE AT CARGO TERMINAL 1

by

Dylan Bravo 5580889

Primary TU Delft Supervisor:	Jaap Vleugel
Secondary TU Delft Supervisor:	Patrick Stokkink
Supervisor KLM Cargo:	Mark van Zijl

Sunday 19th April, 2026

Abstract

Background

Cargo handling is a 24/7 industry, with ground services, air transport, oceanic shipping, and more all happening around the clock. While it is common to have cargo facilities and ports run 24/7 or have them close at night, it is less common to have part of a cargo facility temporarily close at night while operations at other areas remain operational. This setup has not had significant research done and therefore remains unoptimized. Such a facility setup exists at KLM Cargo's facilities at Amsterdam Airport Schiphol, where one of its three cargo terminals temporarily closes with the other two remaining open throughout the night, causing operational inefficiencies.

System analysis

Terminal 1 (T1) at KLM Cargo's primary facilities handles the airline's express shipments (or XPS), but is subject to nightly closures, requiring express products that are delivered at night to temporarily be stored at Terminal 3 (T3), which primarily handles general cargo exports, until the former reopens in the morning. This leads to many inefficiencies and problems including XPS packages frequently being delayed and missing early morning flights due to not being transferred and sorted in time after T1 reopens as well as XPS packages impacting the general cargo operations at T3.

In this project presented by the Business Process and Improvement (BPI) department at KLM Cargo, the XPS package processes were mapped, analyzed, and redesigned in order to optimize the cargo delivery process during the night and eliminate operational inefficiencies and hindrances that exist within the current setup. This was be achieved through a variety of methods, including literature research, data analysis, in-person observations, interviews, and Discrete Event Simulation modeling.

Future requirements analysis

Many stakeholders hold interest in seeing this current process be revitalized and replaced with a new process that will ensure XPS packages are sorted sooner, do not get lost, are prioritized better, and reduce the impact on T3 operations. It is also important that the project satisfy all safety and security requirements, fit within the physical spaces of the terminal floor plan, and fall within the monetary limits of the company.

Design solution

The newly proposed design involves installing automated drop-off machines in the walls of T1 that connect to scanning technology and then subsequently into the sorting machine within T1 directly. This will allow for XPS packages that are cleared and ready to be shipped to directly enter the sorting machine of T1 as opposed to having to wait at T3 until the morning. Additionally, holding bins will allow for some XPS packages to be set aside should the system detect potential safety and security risks that require additional human checks. The primary goal with the new solution is to reduce the average amount of time that XPS packages take from the moment they arrive on the premises to the moment they are inserted into the sorting machine and also remove as many XPS packages from T3 at night as possible.

Results

Discrete event simulation (DES) models were used to replicate both the current and new systems. The model for the new process tested various percentage combinations of XPS packages that were either accepted *directly* or were stored in a temporary storage bin to await *additional checks* in the morning. Data given by KLM Cargo provided values for the interarrival times of the trucks carrying XPS at night as

well as the amounts of packages that would be arriving, allowing for accurate, data-based portrayals of the processes. The simulations provided many outputs for the current and new processes, including average XPS flow times, average times when the first XPS package entered the sorter, and the average number of packages entering the sorter before T1 reopening. These results were compared between the two systems to allow for conclusions and recommendations to be made.

Discussion and recommendations

The results obtained from the DES simulations presented the conclusion that an automated drop-off system can significantly improve the operations and execution of the night-time cargo delivery process of express shipments at T1, showcasing that a minimum of 15% of night-time XPS arrivals being directly accepted into the sorter during the night would be enough to surpass the current system in performance. Additionally, the results supported the notion that the more XPS packages directly inserted into the sorter, the less impact and problems would arise in T3's operations during the night. It is therefore strongly recommended that KLM Cargo continues with development and research that heads toward turning this drop-off proposal into reality.

Acknowledgements

Despite only one name being shown as the author for this thesis, this project would not have been remotely possible without the village of support that I had around me. This thesis is the culmination of nearly five years at TU Delft and there are many people that I'd like to acknowledge for my journey here.

Firstly, I'd like to thank my amazing advisors. From TU Delft, I'd like to thank **Dr. Jaap Vleugel** for making this thesis less daunting, helping calm things when they felt overwhelming, and for always guiding me in the right direction in the most gentle, respectful, and helpful way possible. This really helped keep stress levels down and helped me stay grounded. I also want to thank **Dr. Patrick Stokkink** for being an incredible advisor in the field of coding, modeling, and simulation. These fields have always been challenging for me and it meant so much to me to have such a helpful, patient, and kind mentor in this field.

From KLM Cargo I want to give a huge thank you to **Mark van Zijl** for allowing me to join the KLM Cargo team, for finding this project for me, and for being such a welcoming and helpful face at KLM Cargo. I couldn't have asked for a better industry advisor. Additionally, I'd like to thank all the **coworkers at KLM Cargo** that helped me with my numerous questions, interviews, and data collection. Everyone I met was so kind, respectful, and always willing to help me which was just so nice.

A massive thank you to all my friends that have helped me through the last few years. From **TUD Crew** to the **Delftsche Dodgers**, my **IKEA fam** to **Rampage Army**, **FH2++** to everyone else in between here in NL, you guys made my time in Delft as incredible as it was. Big thank you to all my **friends back in America and Canada** as well. Distance doesn't matter in my appreciation for the support you guys have sent me as well.

I'd like to specifically thank **Susanne**, **Prakhar**, **Nathan**, **Aleksandra**, **Niklas**, and **Floris** for sticking with me through thick and thin, and making my time in Delft honestly one of the most memorable, meaningful, and impactful times of my life. I honestly don't know how I'd have gotten to this point without each and every one of you.

I also want to thank my boyfriend **Roy** for being such an incredible anchor over these last few months. You have been such a rock for me in so many ways and I am so grateful for everything that you have done for me.

I want to thank **everyone in my family** for always, always being there for me and sending me support from all over the world, whether in Canada, England, Nederland, America, or Curacao. I love you all so incredibly much. Huge thank you to **the Kneerims** for somehow always finding a way to be around when I need you guys, you're the best.

I don't think I can even put into words how thankful I am to my parents. **Mama** and **Daddy**, you guys have sacrificed so much and worked so hard for me to be able to even have opportunities like this thesis. I am forever grateful for what you both have done for me and I love you both so incredibly much. **Kyrsten**, **Marieke**, and **Nat** I am also at a loss of words for how much you guys mean to me, I couldn't have done this without you all. I couldn't have asked for better siblings.

Lastly, I want to give my final thanks to **Michelle**. I hope that when you look down from the stars above that you are proud of your little brother and that you know how much of an impact you made on me. I could not have made this journey without you and know that I always carry you around in my heart. Thank you so much for everything, Michelle. Love you and miss you always.

Preface

Express shipping has become an increasingly vital part of modern society, particularly due to the rapid growth of e-commerce over the past few decades. Each year, more and more packages are transported across the globe by air, allowing time-sensitive shipments to reach their destinations more quickly and efficiently. In this industry, time truly equals money, and even small improvements in operational processes can lead to meaningful benefits throughout the express shipment chain.

KLM Cargo is no newcomer when dealing with express shipping and is constantly finding ways to improve their performance and operations. Their very own Business and Process Improvement (BPI) department specializes in identifying processes that will benefit from optimization or redesign, which in turn will benefit the company. They noted inefficiencies in their acceptance process of express shipments during the night and aimed to kick off the process to finding a better solution.

Through a previous project involving improving the truck docking process at KLM Cargo's terminals, I became familiar with the work of the BPI department which was directly in line with what I enjoy doing, being finding obsolete and outdated systems and processes and developing improved and better solutions. When Mark van Zijl from the BPI department presented me with the opportunity to tackle this problem, I was more than eager to accept.

Working on this project pushed me well outside of my comfort zone. It required me to explore new tools and methods, dive deeper into modeling and simulation techniques that were initially unfamiliar to me, and juggle the requirements and perspectives of multiple stakeholders while working within the limitations and constraints present. Although this project proved to be one of the most challenging I have worked on, it was also one of the most rewarding. I am grateful for the opportunity to work with KLM Cargo and to gain insight into the complexities of real-world cargo operations.

Dylan Bravo
Delft, April 2026

Contents

List of Figures

List of Tables

Nomenclature

Abbreviation	Definition
ACN	Air Consignment Note
AMS	Amsterdam, Schiphol Airport
B&B	Build-Up and Breakdown
BPI	Business & Process Improvement
DEP	Departure
DEPNOK	Departure Not Okay
DEPOK	Departure Okay
DG	Dangerous Goods
DIP	Diplomatic mail
DGR	Dangerous Goods Regulations
EPS	Express and Postal Solutions
FoH	Freight on Hand
FWB	Freight Waybill
IATA	International Air Transport Association
KLM	Koninklijke Luchtvaart Maatschappij (Royal Dutch Airlines)
KLM Cargo	Air France - KLM Martinair Cargo
KPI	Key Performance Indicator
LAT	Latest Acceptance Time
LHO	Living human organs
LHP	Local hub performance
RCS	Ready for Carriage (Cargo and documents are received and accepted by airline/handler)
SPL	Schiphol
T1	Terminal 1
T2	Terminal 2
T3	Terminal 3
ULD	Unit Load Device
XPH	Express heavy
XPS	Express shipment

1
Introduction

In an eight month period, a master's student from the Transportation Infrastructure and Logistics program at TU Delft worked on a project from the Business and Process Improvement department at KLM Cargo in order to improve the night-time cargo delivery process of express shipments.

1.1. Context and motivation

The advent of air travel in the early 20th century and the ability to send mail and freight via airplanes changed the way the shipping industry developed, causing increased connectivity, economic growth, and further developed global trade, contributing to the ever increasing globalization that took place during this time. As opposed to ocean shipping which takes weeks of transit time, sending post, cargo, freight, and more via the air continued to become faster, cheaper, and more efficient as technologies improved over the decades. While airplanes cannot match the incredible capacity of ocean vessels, their much higher speeds now became a crucial advantage that could be commercialized, causing a boom in the development of expedited freight and transforming the cargo industry into the extensive global network that it is today.

The air cargo industry is comprised of many coordinated parts and systems including airplanes, cargo terminals and ground facilities, loading and unloading processes, movement of packages, tracking systems, and intense scheduling. Inefficiencies and delays can spring up not only within any one of these parts but also in the relationships between them. It is therefore vital that these processes run as smoothly as possible and can provide the best results possible.

Improving this industry only contributes in bringing far-flung regions and people closer together and aids in facilitating global trade, increasing the availability and distribution of humanitarian aid, and allowing for cheaper and faster access to expedited shipping to more and more people, the same way that larger and more efficient aircraft allowed for air travel to be more affordable and available to the masses. For KLM Cargo, improving and streamlining their processes allows for increased economic gains, a higher satisfaction rate amongst their clients and customers, an increased reputation, increases reliability, and further cements their status as a reliable, safe and efficient carrier.

From a young age, I have been interested in planes and all aspects of commercial air travel. Not only was I enamored by the technologies and machinery that made it possible for these behemoth machines to get off the ground, I also loved learning about how air travel changed the world. Being able to hop on a plane in New York and see family in Europe later that same day or be able to send a present to friends on another continent was really amazing to me and I knew that I wanted to be a part of this industry when I grew up and just do what I could to make it even better.

1.2. Background of KLM Cargo and project

The airline KLM (Koninklijke Luchtvaart Maatschappij) is the Dutch national flag carrier and was founded in 1919, holding the distinction of being the oldest airline in the world still operating under its original name. It operates a wholly owned air freight service named KLM Cargo which deals with all freight that the airline handles. In 2004, the French national flag carrier Air France and KLM merged into one company, the Air France-KLM Group, within which each airline would become a wholly independent subsidiary operating under their respective brands. Additionally, Dutch airline Martinair became fully owned by KLM at the end of 2008 with its passenger services being absorbed into KLM from 2011. From then on, the entire cargo division of the parent company Air France KLM is known as Air France-KLM Martinair Cargo, consisting of subsidiaries KLM Cargo, Air France Cargo, and Martinair, all operating in one unified network with the goal of transporting freight all across the world on behalf of its many clients.

At Amsterdam Airport Schiphol (one of two hubs that Air France-KLM operates from the other being Paris Charles de Gaulle Airport), KLM Cargo operates out of three cargo terminals: Cargo Terminals 1, 2 and 3. The latter two primarily handle the import and export of standard cargo, consisting of europallets and unit load devices (ULDs), while Terminal 1 focuses on express packages, animals, post, and valuables.

Business and Process Improvement is a department within KLM Cargo located in Terminal 1 that focuses on optimizing the companies operations wherever possible, constantly seeking new avenues to explore and develop that will allow for the optimization of KLM Cargo's operations. The project to be undertaken in this thesis was presented by members of BPI as well as the company's Commercial and Business division and aims to eliminate a problematic operational bottleneck in the night-time delivery process of express products to the companies facilities at Schiphol that affects subsequent operations in Terminal 3 and contributes to a variety of inefficiencies and problems.

1.3. Problem definition

In this section of the thesis proposal, the overall problem at hand that KLM Cargo faces is outlined. First, a general overview of the situation and the problem is provided. Next, the overall goals that the company would like this project to achieve are stated. The scope within which the project will remain is then defined. Thereafter, a literature review is performed in which a research gap is identified that the project aims to fill. Lastly, the primary research question and supplementary subquestions that the project will answer in order to achieve the company's goals are listed.

1.3.1. General overview

KLM Cargo's operations are run out of three dedicated cargo terminals at Schiphol Airport, numbered 1, 2, and 3. Terminals 2 (T2) and 3 (T3) work in tandem to handle standardized cargo, namely europallets and unit load devices (ULDs), with one handling imports and the other exports. Terminal 1 (T1) handles more specific cargo, including express packages, post, animals, and valuable cargo. While T2 and T3 operate a 24/7 schedule, T1 does not and closes each night. Despite this temporary closure, cargo deliveries destined for all three terminals continue to arrive throughout the night. To accommodate shipments intended for T1 during these off-hours, incoming freight is temporarily received and stored at T3. Once Terminal 1 reopens in the morning, the cargo that accumulated overnight is shuttled back to Terminal 1 for final processing and sorting. This arrangement ensures that night-time deliveries can continue uninterrupted while still accounting for the temporary closure of T1. A more detailed description of the current state analysis can be found in chapter 4.

1.3.2. Problems faced

The current night-time operating setup and procedures create several problems for KLM Cargo. These issues were identified through interviewing KLM staff and personnel, reviewing data performance statistics, and studying internal KLM documentation. Additionally, they were corroborated with on-site, in-person observations. **It is important to note that the following points describe the problems that arise from the current process setup as opposed to the underlying bottlenecks that cause them, which are explored in section 4.3**. A list of the primary problems experienced is provided below:

1. **Express packages delayed or miss morning flights**
   The largest problem faced is that shipments that are time-sensitive are not being treated as such, as all packages delivered in the night are not being processed and sorted immediately but rather until the morning. Additionally, many packages are sorted too late in the morning once T1 reopens

which causes them to miss their intended flights, inciting further delays.

2. **Express packages inadequately tracked and become temporarily or permanently lost**
   The current method of tracking XPS packages within T3 is very bare-bones and rudimentary. This causes packages to be frequently lost, unaccounted for, misplaced in the wrong section of the terminal, and/or unable to be immediately found should this be required.
3. **Express packages interrupt night-time operations at Terminal 3**
   The mere presence of XPS packages in T3 already presents inefficiencies as it competes for the same physical area, resources, equipment, and workers that are being used for T3 operations. Additionally, the temporarily waiting belly carts that the XPS packages are put into consume floor area that could otherwise be used for the processes in T3. Lastly, extra time and labor is wasted shuttling the accumulated cargo back to T1.
4. **Express packages are not sorted in priority**
   XPS packages delivered at night are all stored together in belly carts regardless of their individual departure times. When they are all shuttled back to T1 in the morning, they are inserted into the sorter one-by-one solely based on handling order. Time-critical packages destined for a flight departing in two hours therefore are treated the same as a package that has a departure time 10 hours from then.
5. **Increase in both vehicular traffic at T3 and long truck waiting times**
   T2 and T3 already face intense levels of vehicular traffic at night for their own operations, especially during peak periods such as Friday nights. Redirecting trucks intended for T1 to T3 further contributes to congestion and extends waiting times for both XPS and general cargo deliveries.

These problems combine together to cause significant issues for KLM Cargo. Frequent delays in XPS packages causes trickling delivery delays and triggers financial consequences, such as a lower throughput of packages, additional handling and rebooking costs, compensation to customers for late or lost deliveries, and subsequently underutilized aircraft and lost revenue. These all also negatively impact KLM Cargo's reliability and reputation in the competitive air-freight market, even causing some customers to overlook KLM Cargo and Schiphol entirely for other airports such as Paris, Brussels, and Frankfurt which have round the clock express package services. Should this not be addressed, the company's strategic positioning will be hindered and it can lose market share.

Problem statement
The previously aforementioned issues can be consolidated into one official problem statement for the project which is stated as follows:

The current night-time closure of T 1 at KLM Cargo requires express shipments to be processed indirectly via T3, which creates significant operational inefficiencies, frequent delays, lost packages, and increased congestion. This setup ultimately causes a lower reliability, lost revenues, increased costs, and a hindered reputation, all of which weakens the company's competitive position.

1.3.3. Project goals and aim

This project was proposed by the Business and Process Improvement (BPI) department of KLM Cargo. This department works to optimize operations wherever it can across the company, operating under the following motto:

"The Business and Process Improvement (BPI) department aims to efficiently and professionally transform and adapt our Cargo organization where necessary. To this end, we implement program, project, and change management for KLM Cargo and AFKLMP Cargo."

BPI identified the problems arising from T1's nightly closures as one that could be improved and wishes to develop a project to determine a solution. BPI defines the following **general goals** any project for the company must fulfill:

1. Increased safety and security
2. Transparency in all processes and real-time information
3. Time-optimized process for efficient operations

4. Cost reduction

In addition to these BPI general goals for any project they oversee for KLM Cargo, this thesis aims to satisfy many **project-specific goals** with any new solution for the night-time cargo delivery process for express packages, these being:

1. Reduce delays and missed flights for XPS packages
2. Improve location tracking and prioritization for XPS packages
3. Mitigate the impacts that handling XPS packages has on T3 operations (both internally within the warehouse and externally with truck congestion)
4. Shorten time and distance that packages and cargo need to travel from delivery point to destination point

All these goals will be achieved through the reduction of the time XPS packages spend in the system, decreasing the number of packages requiring processing at Terminal 3, and maximizing the number of packages having entered the sorter prior to the reopening of Terminal 1 in the morning, all of which are KPIs for the project (see Section 3.2.6 in Chapter 3 for more details on KPIs).

These goals reflect what BPI would like to achieve at the strategic level for the delivery process. See section 5.2 in Chapter 5 for a more in-depth view of the specific objectives and criteria that define how the new design will achieve achieve these project specific goals.

1.3.4. Scope

It is important to clearly identify and differentiate the areas that fall within the project's scope and those that do not. A scope that is too broad can make the project overly complex, prolong timelines, potentially cause unachievable outcomes, and divert attention from solving the core problem. On the other hand, a scope that is too narrow may lead to inadequate results with limited impact, fail to address the critical aspects of the issue, or overlook important dependencies. Establishing a "Goldilocks zone" for the scope is therefore crucial to ensure the problem is fully addressed while keeping the project manageable and achievable.

Physical scope

The project will focus on the floor area of T1 and its immediate surroundings, as this is the final destination for the cargo in question and the location where the redesigned delivery process will be implemented. Because temporary night-time storage currently takes place in T3 and affects its operations, T3 is also included in the physical scope. The existing physical connection between T1 and T3 will likewise be considered, as it is directly relevant to the current process this project aims to improve.

Although T2 is physically connected to and works in tandem with T3, it primarily handles cargo arriving from airside operations and plays only a minor role in the night-time delivery process and thus is far less affected by XPS operations. The transfer process that XPS packages take from T3 to T1 navigates through T2 utilizing their interior pathways, and some transito XPS packages are broken down and stored in T2 (see 4.2.2 for more detail). However, these interactions do not have as drastic an impact as those observed in T3. For these reasons, T2 and its operations are acknowledged within the system context, but are primarily excluded from the scope for this project.

Operational scope

This project focuses exclusively on the operations of the KLM Cargo terminals during the night. Processes during the day, when all terminals are open and operate normally, are excluded, as the primary issues arise during the nightly closure of T1. The disruptions occur when cargo destined for T1 must be temporarily stored in T3 at night, creating operational inefficiencies that do not exist during daytime operations.

Additionally, the project will examine scheduled, recurring operational patterns, specifically the fixed alternation between daytime (all terminals open) and nighttime (T1 closed), rather than unexpected, uncommon, or extraordinary events (pandemics, accidents, etc.). While in these cases, having research on alternating schedules with temporary terminal closures can be applied to these scenarios where suddenly, a terminal may be rendered unusable or different procedures need to be established for cargo delivery, this study is not designed to address these types of scenarios and is specifically geared to addressing issues at terminals that have structured, planned systems with predictable closures, not developing contingency plans for rare emergencies.

Future developments
It is important to note that all KLM Cargo operations are to be moved to a brand new cargo terminal due to open in 2030 at the earliest. This terminal will consolidate all operations into one building and introduces many new technologies, systems, protocols, and procedures. While information and research performed in this report could theoretically and potentially be useful in the planning and development of the new terminal, for the purposes of this project, the new terminal will remain outside the scope, and its influence will not play a role in the research, design, and results of the solution ultimately presented in this thesis project.

1.4. Research Questions

To address the problems KLM Cargo faces with the current setup, the following primary research question and subsequent subquestions to be answered by this thesis report were developed.

1.4.1. Primary research question

The main question that this thesis aims to answer is as follows:

PRIMARY RESEARCH QUESTION

How can the delivery process of express packages during nightly closures of T1 be redesigned to optimize flow, improve operational reliability, and minimize transport time and distance, while complying with relevant safety and stakeholder requirements?

1.4.2. Subquestions

To aid in the answering of this primary research question, six subquestions were formulated that help with the collection, analysis, and evaluation of relevant data and information. These are as follows:

1. What are the size, layout, and capacity constraints of the physical spaces for the current processes and the proposed design and what are the quantities, types, physical characteristics, and storage patterns of cargo being delivered?
2. What are the sequential steps of and bottlenecks present in the current night-time delivery process, and which KPIs are appropriate to best evaluate and compare both the current and proposed processes?
3. What regulatory, safety, security, and stakeholder requirements must the new delivery system meet, and what relevant technologies, systems, or operational practices from other companies or industries could be incorporated into the final design?
4. How can the nightly cargo delivery process be designed efficiently with the information obtained through the previous sub-questions taken into account in order to achieve the highest possible improvements?
5. How can the proposed new design for the nightly cargo delivery process be best represented theoretically as a model?
6. How can this theoretical model be simulated and used to evaluate and compare the performance of the proposed design against the current state?

1.5. Thesis structure

This report takes the form of a thesis project. Chapter 1 introduces the problem at hand that KLM Cargo is facing and identifies the gap in scientific literature that this research aims to fill. Chapter 2 provides the literature review undertaken which established a research gap that the project aims to fill. Chapter 3 details the methodology performed in order to gain the necessary information, data, and criteria needed to map the current process and develop the new solution design. Chapter 4 then presents the current situation and how the delivery system presently is performed. Chapter 5 delves into the relevant stakeholders and their respective stakes, requirements, and desired objectives. Chapter 6 presents the proposed new solution in full, both in a physical aspect and a procedural aspect. Chapter 7 provides the model and simulation that were developed to replicate the current and new processes in a computerized manner while Chapter 8 presents the results from these simulations. Chapter 9 discusses these results while Chapter 10 concludes the report by presenting conclusions based off these discussions and listing recommendations to KLM Cargo and potential areas for future research.

See below a visual diagram of how the thesis is structured:

	SUBQUESTION	CHAPTER	METHOD	OUTPUT
PROJECT PHASE 1	Subquestion 1	4	Literary research and benchmarking, Document and written procedure analysis, Data processing and analysis, Interviews, Observations and physical visits	Physical characteristics and constraints of current process
	Subquestion 2	4	Literary research and benchmarking, Document and written procedure analysis, Data processing and analysis, Interviews, Observations and physical visits, KPI establishment	Current procedures and process mapped
	Subquestion 3	5	Document and written procedure analysis, Interviews	Full requirements and criteria list
PROJECT PHASE 2	Subquestion 4	6	Interviews, KPI establishment, Brainstorming method	Complete physical design and written procedures for new process
	Subquestion 5	7	Data processing and analysis, KPI establishment, Mathematical models and simulations	Fully functioning model replicating current and new process
	Subquestion 6	7	Data processing and analysis, KPI establishment, Mathematical models and simulations	Quantitative results from simulations

Figure 1.1: Diagram for thesis structure

2 Literature review

An important step in the thesis proposal is performing an comprehensive literature review. This involves combing through academic papers and scientific research to get a solid understanding for what areas of a topic are already known and have been extensively examined as well as what areas have inconsistencies and other shortcomings and require further testing and research.

In this section, a background on the air cargo industry and night-time terminal operations is provided. Next, a search strategy for the literary review is presented that was followed in order to establish the areas of the topic that have been covered and where knowledge gaps still remain. These findings are then detailed in an analysis section, after which a definitive knowledge gap is determined in the conclusion section. In the final section, the importance and relevance of addressing and filling this knowledge gap is discussed.

2.1. Background

The air cargo industry is a key component of the global logistics network, connecting goods across continents through a system of airports, terminals, and hubs. Efficiency is critical, as it directly impacts delivery speed, operational costs, and supply chain reliability. The cargo handling process involves multiple complex steps, requiring adequate facilities and infrastructure to function effectively. Cargo terminals serve as central nodes where freight is processed, sorted, transferred, or temporarily stored as it moves between landside and airside operations or continues to its destination. The effective management of these terminal activities is therefore essential to maintaining a reliable and efficient air cargo process, making terminals a key focus for research and optimization.

As air cargo is a global operation involving countries and cities across all time zones, many operations run around the clock, including night-time hours. Night operations are crucial for meeting delivery schedules, maximizing terminal throughput, and ensuring timely connections with flights. However, operational constraints during night-time can create bottlenecks and temporary challenges that are not present during day-time operations. Understanding how night-time processes interact with daytime operations is essential for designing an optimized cargo delivery system.

With the increasing digitalization occurring throughout cargo industries, the use of artificial intelligence and automated systems are becoming more prominent within these networks. This necessitates in-depth analysis of their effects (both beneficial and detrimental) and understanding how automation and AI will play a role in future improvements as well as the concerns relating to the removal of human oversight in traditionally manual roles.

In summary, the global air cargo network relies on highly coordinated, continuous operations involving a variety of cargo types and complex terminal processes. Understanding these operations, particularly during night-time hours when constraints or operational challenges may occur, is critical for designing efficient cargo delivery systems. It is therefore vital to continue researching methods and ways of improving cargo terminals and their operations, particularly at night, in order to optimize cargo flow, minimize transport inefficiencies, and ensure operational safety and compliance.

2.2. Search strategy

In a literature review, scientific papers and academic journals will be systematically examined in order to identify areas where extensive research has been conducted and, more importantly, gaps where research has not adequately provided analysis, insights, or new information. To perform a literature review based on the background of air cargo transport and night-time cargo terminal operations, a search strategy is implemented. In this strategy, multiple concept groups are created, which serve as thematic categories for organizing the literature. Within these concept groups, a variety of keywords are defined, each representing specific terms or topics relevant to the research focus. Finally, a truncation strategy is designed by combining the concept groups and their associated keywords, which will form the basis for searching academic databases and repositories such as Google Scholar and Scopus. The conceptual and methodological framework for the literature review employed in this research project is presented in Table 2.1:

Concept groups	Location; Cargo type; Operations; Variables; Time; Technology/Automation	
Keywords	*Location:*	cargo terminal*, port*, airport*, airside, hub*, warehouse*, facilit*, yard*, "distribution center*"
	Cargo type:	carg*, "air cargo*", freight*, packag*, pallet*, ULD*
	Operations:	logist*, operat*, process*, deliver*, transport*, transfer*, stor*, hold*, shipment*, "drop off*"
	Variables:	efficien*, delay*, distance*, cost*, time*, throughput*, turnaround*, productivit*
	Time:	night*, "night operation*", schedul*, 24/7, "temporary closure*", curfew*, off-peak*, shift*
	Technology/ Automation:	automation*, automat*, robotic*, autonomous*, AI, "artificial intelligence", "machine learning", scanner*, "X-ray", screening, "self-service", locker*, "automated drop-off", digital*, technolog*
Truncation	(Location) AND (Cargo type) AND (Operations) AND (Variables) AND (Time) AND (Technology/Automation)	

Table 2.1: Conceptual and methodological framework for literature review

This framework was applied across multiple search engines, yielding many results. Articles deemed most relevant to the research topic were selected for further analysis. For each useful article, two additional methods were employed to identify related journals and sources: *forward snowballing* and *backward snowballing*. Forward snowballing involves examining all the papers that have cited the original article (for example, Paethrangsi (2025) [22] was found by forward snowballing the paper Nath & Upadhyay (2024) [20]), while backward snowballing entails reviewing the references that the article itself cites (for example, Chao (2013) [6] was determined by looking at the citations of the paper Hong (2025) [10]). By using both approaches, additional relevant studies within the same thematic area could be identified, providing valuable information to support the literature review and contribute to addressing the research question.

2.3. Analysis of themes

The papers analyzed for this literature review covered multiple aspects of the air cargo industry as well as methods and strategies to make operations more efficient and optimized. From these studies, multiple themes emerged, with six standing out as the most relevant to the primary problem faced at KLM Cargo. Each of the six relevant themes contribute unique insights into how air cargo terminals can improve efficiency, reliability, and adaptability. The most common themes explored, along with the papers that best relate to them, are outlined below:

2.3.1. Improving air cargo operations

The papers of Nath & Upadhyay (2024) [20], Lapphiphitmongkhon (2008) [19], and Paethrangsi (2025) [22] focus on the handling and optimization of cargo operations at air terminals in specific airports, namely in India, Taiwan, and Thailand respectively. These studies investigate factors such as workforce management, process sequencing, determining key bottlenecks that affect operations, and equipment utilization, demonstrating that even small changes in operational procedures can have significant impacts on throughput and turnaround times.

Similarly, Wei (2005) [36] and Selinka (2016) [25] adopt a broader perspective on improving terminal performance. Wei emphasizes identifying and targeting key bottlenecks within complex processes in order to optimize processes, while Selinka applies time-dependent performance approximation to truck handling operations, offering a framework for predicting congestion and dynamically allocating resources. Together, these papers highlight a variety of strategies ranging from procedural changes to predictive modeling that can be applied to optimize the general operations of air cargo terminals.

2.3.2. Time-specific cargo operations

Several papers addressed the time scheduling aspects of cargo handling, where demand patterns, staffing levels, and throughput vary throughout the day or night. Gurturk (2025) [9] focuses specifically on night-time cargo operations, identifying unique challenges such as limited workforce availability, noise restrictions, and stricter cut-off times for departures. Thein (2025) [31] examines 24/7 terminal operations in a SkyCargo facility, providing insights into how continuous schedules can be managed to reduce downtime and balance day/night workloads. These papers show that air cargo operations are highly sensitive to time-dependent demand, and they stress the importance of flexible scheduling, demand forecasting, and real-time decision support to maintain efficiency under varying scheduling conditions.

2.3.3. Facility layouts

The physical design and spatial organization of cargo terminals were key themes in the works of Priansyah et al. (2024a, 2024b) [24] and Ballis (2007) [2]. The first two papers present case studies of Indonesian and Chinese airports, demonstrating how to incorporate requirements and constraints into a new design and comparing layouts of logistics parks that will best benefit the airports' respective cargo operations. Ballis takes a more general approach, providing a framework for evaluating cargo terminal layouts that balances space constraints with process efficiency. Collectively, these studies emphasize that facility design is not merely a construction issue but a dynamic operational factor that directly affects throughput, handling times, and the scalability of cargo terminals.

2.3.4. Tools and modeling

A number of papers explored the use of analytical tools and mathematical models to evaluate and optimize terminal operations. Tubis (2024) [32] introduces augmented reality as a decision-support tool to visualize cargo flows and identify process inefficiencies. Hong (2025) [10] applies structural equation modeling to examine agility and competitiveness in air cargo networks, while Chao (2013) [6] employs quantitative evaluation models to assess operational performance and compare alternative strategies. These approaches provide robust, data-driven methods for testing scenarios and forecasting outcomes, illustrating how advanced modeling can guide both short-term decisions and long-term infrastructure planning.

Additionally, the paper of Ballis (2007) [2], previously referenced when discussing facility layouts, also touches upon modeling, by presenting an analytical method for determining the appropriate size of storage facilities with minimal computational effort, and by referencing modeling approaches that are primarily simulation-based that enable a more detailed examination of various air cargo terminal design aspects

2.3.5. Automated baggage handling and self-serve drop-off technology

This theme is the first of two focusing on the use of automation and the application of advanced digital technologies and artificial intelligence in a cargo industry setting. The first three papers discussed all focus on different aspects of automated baggage handling systems. The paper by Papoutsidakis et al.

(2019) [23] presents a case study examining the implementation of automated baggage handling systems in modern transportation. It discusses the financial and operational efficiency benefits that these newer systems bring, especially as they become cheaper to acquire and are more accessible. Similary, Singh (2023) [27] explores innovative baggage handling solutions designed to enhance passenger experience, highlighting how automation can reduce manual handling, increase processing speed, and improve overall system reliability.

The report by Kleshko et al. (2024) [12], which investigates automated baggage screening and logistics systems for improving airport efficiency, also identifies several operational benefits of these technologies. These include increased throughput, reduced human error, improved tracking of individual items, and more streamlined coordination between different stages of baggage processing. Collectively, these studies emphasize how automated handling systems can significantly improve operational efficiency while reducing dependence on manual labor.

It is important to note that in all of these papers, the focus was on passenger baggage systems. While that is not the focus for this project and for KLM Cargo, the machinery and underlying systems are largely comparable with only the type of item being transported varying. Therefore, this technology is very applicable when considering introducing and implementing automation in an air cargo setting.

On a similar note, the papers of Shah et al. (2024) [26] and Thakur et al. (2025) [30] provide in-depth analyses of automation for passenger baggage as well, but focus specifically on automated drop-off systems. These systems allow passengers to independently check in baggage using self-service machines that scan boarding passes, weigh and label baggage, and automatically route items into the sorting system. These papers report that airport operations were enhanced through these automated baggage drop-off and boarding pass generation systems by allowing for reduced processing times, higher throughput of items, and and less overall congestion in the airport, clearly indicating that similar systems could provide similar benefits to air cargo services still relying on a lot of manual labor.

Together, these papers show there clearly exists an extensive amount of research on the burgeoning development of automation due to the financial, operational, and reliability benefits. They support the notion that automation can significantly improve handling efficiency, reduce manual workload, and increase system throughput in logistics environments. However, while there exists a lot of research on this topic independently and focusing on standard airport settings, there is no mention or application of systems in airport terminals or facilities with varying day night schedules and partial terminal closures which would've provided additional, crucial information for this topic.

2.3.6. Security screening with artificial intelligence

The second of two themes focusing on the implementation of artificial intelligence and automation shifts attention to utilizing these technologies for security and protection purposes. In this context, digital systems are increasingly able to detect security threats, dangerous goods, tampered materials, and illegal items within bags, packages, and cargo.

The papers by Kleshko et al. (2024) [12] and Vukadinovic et al. (2022) [35] focus on using AI in baggage screening and X-ray inspection services. These studies analyze how machine learning algorithms and automated image recognition technologies can help in identifying potentially dangerous items more quickly and accurately than manual screening methods. They indicate how these systems were found to be reliable, accurate, and efficient which greatly aided in threat detection, efficiency of the security lines, and reducing overall costs.

Turevcek et al. (2025) [33] similarly investigates the use of AI in airport security screening, but from more broad perspective. They explore how artificial narrow intelligence (ANI) which is on machine learning can be integrated across varying screening technologies such as imaging systems, metal detectors, and spectrometric detection methods to improve threat detection. Additionally, impact of these technologies on screening capacity using a queueing framework based on Jackson Networks is analyzed. The results from this paper support the notion that AI screening can improve both processing efficiency and detection reliability while reducing the time spent at security checkpoints.

Lastly, the reports by Okolo et al. (2022) [21] and Blakenburg et al. (2026) [3] shift the focus away from passenger baggage to general cargo screening across borders as well as customs controls. The former zeroes in on intelligence border security and the latter specifically with the flow of defense and dual-use goods. These papers and their contents more closely match the type of items and cargo that KLM Cargo checks and thus their results and conclusions are incredibly relevant and important to analyze.

Similarly with the results from the previous theme, all of these papers provided supporting evidence that these automated systems are vital to take into account in any future air cargo procedures and systems, but do not specifically examine their application in cargo facilities operating under varying schedules or partial terminal closures. Thus, there still exists some research missing that would have been helpful for KLM Cargo.

2.4. Conclusion of the literature review

Taken together, these six themes reveal a large body of research dedicated to improving the speed, cost-effectiveness, reliability, and technological capabilities of air cargo handling. However, while they cover a broad range of operational, spatial, and technological factors—including process optimization, facility design, automation, and security technologies—none directly address the unique challenge of facilities where certain terminals temporarily close while others remain active.

Despite the many existing themes of research on air cargo terminal operations, several notable knowledge gaps remain. Topics such as alternating day/night operational schedules, full cargo facility closures, and the application of specific analytical methods (e.g., Markov models or queuing theory) to air cargo operations have received limited to no attention. In particular, there is a striking lack of research on situations in which *some terminals in multi-terminal cargo facilities temporarily close at night while others in the same facility remain fully operational*.

This gap is especially relevant to KLM Cargo's current operational procedures, where Terminal 1 shuts down during night hours while other terminals continue to operate around the clock. Such a setup creates fluctuating capacity and demands adaptive procedures to manage cargo flows, staff allocation, and vehicle movements as operations transition between full-capacity and partial-capacity states.

Addressing this gap offers significant practical and scientific value. By developing official strategies and validated models for managing operations during partial terminal closures, airports with similar setups could reduce delays, prevent bottlenecks, and improve cost efficiency. Furthermore, such research would provide a foundation for future terminal design, enabling planners to incorporate flexible closure strategies into new facilities and retrofits. Insights from modeling approaches such as queuing theory, Markov models, and discrete event simulation could play a central role in testing alternative layouts, predicting system behavior under closure scenarios, and establishing key performance indicators (KPIs) to guide decision-making.

2.4.1. Gap definition

Despite considerable research on improving standard cargo terminal operations, time-specific cargo operations, air cargo facility layouts, the use of some mathematical tools and modeling in freight situations, and the application of automated systems and artificial intelligence, there remains a notable absence of focused analysis on cargo facilities in which certain terminals close temporarily while others remain continuously operational. This absence leaves an opportunity to develop new knowledge, data, and operational procedures that can benefit both the cargo industry and the broader scientific community.

The official gap definition, therefore, is as follows:

GAP DEFINITION

A notable absence exists of focused analysis on cargo facilities where some terminals close temporarily while others remain continuously operational.

2.5. Discussion of the literature review

In this section, the importance of this research for both current and future scenarios, situations, and locations is explained, before the steps to be taken in order to fill the aforementioned gap will be detailed.

2.5.1. Research significance

The research obtained from filling this gap will be applicable in many airports and other cargo facilities around the world which currently face similar operational schedules, temporary closures, or fluctuating terminal capacity. At present, most airports facing similar conditions to those seen at KLM Cargo's facilities, such as those in Los Angeles, Frankfurt, and Shanghai [8], rely on improvised and case-specific solutions such as temporary storage or manual rerouting, with no prior research studies or reports to reference. This can lead to inefficiencies and increased costs, which could be mitigated through the development of researched and standardized procedures. These locations would be able to benefit from evidenced-based findings and validated recommendations designed to improve operational performance under conditions similar to those they currently face.

2.5.2. How gap will be filled

The identification of a research gap regarding systematic, temporary night-time closures at air cargo terminal facilities presents both an interesting and highly relevant area of study. This gap is important because, while night-time operations at terminals are common in the global air cargo industry, there is limited research analyzing the effects of structured, scheduled closures on cargo flow, operational efficiency, and resource utilization. By addressing this gap, terminals around the world can benefit from evidence-based guidelines and frameworks for managing cargo during these periods, leading to more efficient operations, reduced bottlenecks, and potentially lower operational costs.

Filling this gap allows the development of processes that are no longer improvised or handled on a case-by-case basis, but rather are informed by systematic research, modeling, and analysis. New designs and procedures can therefore be created using validated methods, improving process reliability and performance. Additionally, insights gained from studying temporary closures can inform the design of future cargo terminals, allowing them to be planned with these operational scenarios in mind. This can lead to more economical, flexible, and scalable terminal designs, a benefit currently limited by the lack of research in this area.

From the literature, several methods and theories appear particularly relevant for addressing this gap. Discrete event simulation and mathematical modeling have been widely applied in cargo operations research to analyze flow, identify bottlenecks, and optimize resource allocation. By combining these approaches, it becomes possible to create robust, validated strategies for managing night-time closures that improve efficiency and stakeholder satisfaction simultaneously.

3
Methodology

This section breaks down the approach plan and methodology for the project. First, the project is broken down into two primary phases with the subquestions split between them based on their respective objectives. Then, the methods to be used in the project for research, data collection, and design development are identified and broken down per method with emphasis placed on which subquestions each method helped answer.

3.1. Project phases

Answering the primary research question requires a wide range of information gathering, research, and analysis and is a complex and arduous process. Thus, breaking down the question aids in creating a clear and coherent research structure and process flow. Therefore, two primary project phases are established with the six subquestions divided evenly amongst them in order to create a more directional flow in how to tackle the main problem of the project and answer the primary research question.

3.1.1. Phase 1 - Current process mapping and analysis

Phase one of the project involves capturing the current process, mapping existing facilities and operations, and analyzing the relevant procedures in order to gain a clear picture of how the delivery process of express shipments presently is organized and run. Additionally, the overall requirements and criteria that a new design would need to take into account are also then determined and defined, allowing for the development of a brainstormed version of the proposed solution design.

3.1.2. Phase 2 - Design development and validation

Phase two of the project involves taking the brainstormed version of the solution design from the previous phase, expanding upon it and refining it in detail, converting it into a theoretical model, running it through a simulation, and validating it through comparison with the current process, after which quantitative results can be obtained and analyzed.

3.1.3. Project phases summary table

These phases subquestions are identified, broken down, and described below in table 3.1:

Phase	Title	Subquestion	Description
Phase 1	Current process mapping and analysis	Subquestions 1, 2, & 3	These subquestions focus on recording the current process and objectively mapping the existing facilities and operations. They involve analyzing this baseline system and defining the requirements that the solution design must satisfy
Phase 2	Solution design selection and validation	Subquestions 4, 5, & 6	These subquestions involve the development of the solution design from the inputs obtained via the methods outlined in section 3.2.1, its expansion in detail, and its validation and comparison to the currently existing process

Table 3.1: Phasal breakdown of the project

Each phase targets specific subquestions and collectively ensures a structured approach to answering the main research question, guiding the project from understanding the current processes, determining stakeholder requirements, proposing a solution design, and modeling and simulating this proposed design. Throughout the rest of the report, color-coded text boxes (matching the color of their respective phase) are used to indicate where each subquestion is addressed and answered.

3.2. Methodology

There are several methods for information gathering and data collection that were utilized for this thesis project. The primary methods, their objectives, and their results are outlined below:

3.2.1. Literary research and benchmarking

To establish a solid foundation for this study, relevant literature was reviewed, analyzed, and synthesized. While Chapter 2 provides a structured review of academic literature to establish the research gap, this section focuses on the use of literature and benchmarking as a methodological tool. Here, external sources are used to extract practical design inputs, technological capabilities, and best practices that inform the development of the proposed solution. This literature is comprised of external research papers, academic journals, relevant online sources, and external company websites and documents. The purpose of this is to identify existing knowledge, best practices, and frameworks that can inform the design of the night-time cargo delivery process.

Additionally, the processes of other industries, companies, and organizations were benchmarked to identify physical implementations of similar systems and designs in practice. These included, but were not limited to, sorting system providers such as Vanderlande Systems and Lödige, as well as large-scale logistics operators including Amazon and DHL, which employ advanced automation and handling technologies relevant to this project. This benchmarking provided insights into feasible solutions and practical considerations that can be applied to KLM Cargo's context.

This aided in answering **subquestions 1 and 2** by providing information on the types of cargo that are bring handled with their relative characteristics, the layout and space usage of the terminals through blueprints and schematics, and process manuals detailing cargo handling procedures and operations.

3.2.2. Document and written procedure analysis

This process is similar to literary research and benchmarking, however it focuses on the analysis and study of internal KLM Cargo files, documents, and materials. The purpose of this is to establish what preexisting knowledge, practices, and frameworks are already present within KLM Cargo. This allows for the generation of a clearer picture of the current processes, where potential bottlenecks are that are contributing to the problems outlined in section 1.3, and areas that can be improved through a new proposed design process.

KLM Cargo documents that were reviewed include the following:

- **Cargo handling manual (CHM)**
  The CHM is a comprehensive guide that "contains the procedures and instructions for the handling of Cargo and Mail" [14]. Three specific chapters were utilized for this report, the first being the very first chapter which outlines the use and reason for the CHM. Chapter 8 which covers how to accept shipments as 'Ready-for-Carriage' [13] and Chapter 16 which covers protocols for general express shipments as well as some with additional special handling codes [15].
- **Werkinstructies / Operationele Instructies / Lokale Procedures Schiphol**
  KLM Cargo provides its workers with work guides and written instructions for a plethora of procedures within the entire company's operations. These documents first list for which task the instructions are for, who the instructions are for, what equipment is needed for these instructions, before outlining in detail a clear step-by-step guide to the instructions. It then discusses safety considerations, who to contact should help be required, and who is in charge of these specific work instructions. These include [34], [7], [29], [5], [4], and [28].
- **Product portfolio 2025**
  The product portfolio details the kinds of transport options that KLM Cargo offers and the conditions required for each of these offers. This was used to obtain more information on the requirements for standard express packages and express heavy packages [1].
- **Terminal blueprints**
  The work floor blueprints for all three terminals (terminal 1 [17] and terminals 2/3 [18] were provided which aided greatly in the physical mapping of the current processes as well as the planning of the physical aspects of the newly proposed solution design.

These documents will help in answering **subquestions 1, 2, and 3** by providing details on current processes, physical spaces, express package characteristics, and operational requirements.

3.2.3. Data processing and analysis

KLM Cargo provided extensive datasets and spreadsheets containing a variety of operational information, including truck arrival, docking, and waiting times, the number of XPS packages per truck, and the count, weight, and volume of the packages [16]. These data were examined, analyzed, and processed to quantify the performance of the current process, identify operational patterns, and support the definition of key requirements and constraints for the new design. From this analysis, additional metrics were derived, including the number of trucks arriving per day, intervals between truck arrivals, and histograms of truck arrivals and packages per truck. Additionally, the datasets provide important input parameters for the models developed to replicate both the current and proposed delivery procedures as closely as possible. They also serve as reference data for the verification of the models and the evaluation of their results.

This analysis aids in answering **subquestions 1 and 2** by providing numerical values and quantitative measurements for the physical spaces, the time and distance that cargo travels, and the characteristics of the cargo itself. These factors are crucial for accurately mapping the existing processes and ensuring that the models represent the real-world system as closely as possible. Additionally, these help with **subquestions 5 and 6** in the construction and validation of the model in code.

3.2.4. Interviews

Multiple interviews with relevant and related individuals were conducted in order to gain valuable first-hand accounts, opinions, experiences, suggestions, perspectives, and desires regarding the current delivery process. These individuals were able to describe both the positive and negative characteristics of the existing system, highlight aspects they believe should be changed, and propose ideas that could contribute to the development of a new design. Incorporating these perspectives was essential, as these individuals interact with the process on a daily basis (both directly and indirectly) and therefore possess detailed knowledge of its benefits, shortcomings, and operational bottlenecks. This type of information cannot be obtained solely through document analysis or industry research.

The interviews explored opinions on current practices, desired improvements, and ideas that could enhance the design of the nightly delivery process. Different interview approaches, such as structured and semi-structured interviews, were used depending on the role of the interviewee and their level of involvement in the operational process.

Specific parties at KLM Cargo that were targeted for interviews include the following:

- BPI management
- EPS department
- ImpEx department
- Warehouse personnel
- Security and Customs department

In total, nine interviews ranging from 30 to 60 minutes were conducted with personnel representing these departments. Each interview was transcribed and subsequently condensed into summarized, anonymous accounts to extract the most relevant insights while maintaining confidentiality. To further preserve anonymity, simplified and generic job titles were used in place of the interviewees' actual job titles in order to prevent accidental identification. Additionally, prior to conducting the interviews, ethical approval was obtained through the relevant TU Delft review process. Consent was obtained from all interview participants, and the resulting transcripts and summaries were stored securely within TU Delft's protected data storage environment.

By interviewing these individuals, vital first-hand insights into cargo flow processes, operational bottlenecks, and potential areas for improvement were obtained. These insights contributed significantly to answering **subquestions 1, 2, and 3** as well as **subquestion 4** and were particularly important for defining the requirements and objectives that guided the development of the proposed design.

The anonymous summaries can all be found in the Appendix from sections D.1.1 to D.1.9 .

3.2.5. Observations and physical visits

In addition to interacting with and interviewing people who have had personal experience with the process, first-hand observations will also be conducted in order to personally see how the process works and get a better picture of the reality that is to be changed with this project. Relying on documents alone only gives part of the picture; observing the process firsthand is essential to fully understand how it works and the challenges this project aims to address. This will aid in identifying the steps of the entire process as well as observing the characteristics of the terminals and the cargo which will help answer **subquestions 1 and 2**.

Observations that were performed include:

- Personal/individual tour of T1 and sorter from shift leader
- Group tour of T2 and T3
- General observations of T1, T2, and T3 facilities and processes during the day
- Night-time observation of the relevant processes executed
- Guided night-time walkthrough of the delivery and transfer processes before, during, and after Terminal 1 reopening by KLM Cargo shift leader

It is important to note that these observations were not conducted as a formal data collection method. Their purpose was primarily to allow the researcher to gain a clearer understanding of the operational environment and to visualize how the processes described in interviews and documents take place in practice. The observations therefore served as contextual support rather than as a source of empirical evidence. They helped verify and better interpret the information obtained through other methods, particularly the interviews, but no systematic data collection, measurements, or detailed observational records were taken. Instead, the visits were intended to provide a general understanding of the facilities, processes, and working conditions in order to better contextualize the findings obtained from the primary research methods.

3.2.6. Key Performance Indicator (KPI) establishment

Once the current processes and system constraints have been mapped and analyzed, appropriate Key Performance Indicators (KPIs) are established to evaluate both the current process and the proposed design. These KPIs provide measurable criteria for assessing operational performance and determining whether the proposed solution improves upon the existing system.

Two categories of indicators are used. The first consists of **static** process metrics that describe structural characteristics of the delivery process, such as total cargo travel distance and physical space utilization. These values are derived directly from process mapping and facility layouts and remain constant once the system design is defined. The second category consists of **dynamic** performance KPIs that are obtained through modeling and simulating the current and proposed processes. These indicators measure operational performance, including metrics such as package time within the system, truck waiting times, and system throughput.

KPI selection process

The KPIs selected for this project were determined based on a combination of factors including the objectives, desires, and operational concerns obtained from the interviewed KLM Cargo personnel, the available datasets provided by the company, and the capabilities of the modeling and simulation methods used. Among these factors, the former two have the greatest influence on the final KPI selection.

Static process metrics

Several static measurements that do not vary between simulation iterations can be directly compared between the current process and the proposed solution design. These metrics describe structural characteristics of the delivery process and are derived from process mapping and facility layouts. The static metrics selected for this project are the following:

- Total distance cargo travels
- Floor area used

Dynamic performance KPIs

The dynamic KPIs represent operational performance indicators that vary depending on system conditions and are determined through simulation of the modeled processes. These KPIs provide insight into how efficiently the system operates under different conditions and allow for direct comparison between the current and proposed designs. The dynamic KPIs selected for this project are:

- Max, Min, Average, and N95 time an XPS package spends in the system
- Truck waiting time
- Number of packages removed from Terminal 3 processing
- Maximum waiting queue length

KPI relevance to project objectives

Establishing these KPIs enables an objective comparison between the current process and the proposed design and directly contributes to answering **subquestion 2**, which focuses on identifying appropriate metrics for evaluating cargo delivery performance. Additionally, the usage of KPIs is vital in answering **subquestions 4, 5, and 6** as these involve developing, writing, and executing the code for the model and simulation which revolves around the established KPIs.

3.2.7. Brainstorming sessions

The brainstorming stage uses the information, data, and resources obtained from the previous methods including literature research, process mapping, interviews, KPI analysis, stakeholder input, and benchmarking studies to begin creating a potential solution design that will in the end satisfy all the necessary requirements, criteria, and objectives.

For this project, the brainstorming sessions were applied to develop both a physical breakdown as well as a procedural breakdown for the redesign for the night-time delivery process for XPS products.

Physical breakdown
To structure the concept development process, the proposed system was decomposed into three primary components, these being the *drop-off interface*, the *internal handling and screening* section, and the *integration with existing infrastructure*. This breakdown allowed each part of the system to be analyzed individually while ensuring that all functional aspects of the design were considered.

Procedural breakdown
In addition to the physical system structure, different operational scenarios were explored to determine how the system would respond to various types of deliveries. These scenarios include both *standard drop-off procedures* and *non-standard drop-off procedures* which consist of non-acceptable packages, packages requiring additional screening, and packages delivered too early for immediate processing. Additionally, the *morning handover process*, *fallback procedures*, and *day-time operational procedures* were also included with new processes brainstormed for each as well.

The brainstorming sessions were used primarily to answer **subquestion 4**.

3.2.8. Mathematical models and simulations

To evaluate and validate the performance of both the current process and the proposed design before implementation, a computational modeling approach was used. Mathematical and simulation-based techniques were applied to model the delivery systems, allowing the two processes to be tested, optimized, and objectively compared using the previously defined key performance indicators (KPIs). This step is essential before implementing any new system in a real-world operational environment.

Discrete Event Simulation (DES)
Discrete Event Simulation (DES) was used as the primary modeling method for representing the night-time cargo delivery processes. DES is well suited for logistics and terminal operations because it models systems as sequences of discrete events over time, such as truck arrivals, package handling, scanning operations, and transfers between terminal areas. This approach allowed both the current and proposed systems to be simulated under realistic conditions, including varying truck arrival patterns and cargo volumes, enabling the evaluation of dynamic KPIs such as package time within the system, truck waiting times, queue lengths, and overall system throughput.

Implementation in Python
The simulation models were implemented programmatically using Python and executed within Visual Studio Code. Python was used to represent system entities, resources, and processes, allowing the behavior of trucks, packages, handling equipment, and terminal operations to be simulated over time. Running the simulations programmatically also allowed multiple iterations of the model to be executed efficiently in order to obtain stable performance estimates for the selected KPIs.

Queuing theory system behavior
Queue-based system behavior is inherently represented within the discrete event simulation. Certain resources in the modeled system, such as loading docks and scanning stations, have limited capacity, which can create queues when demand exceeds availability. For example, trucks may need to wait for an available docking position, while packages may queue for handling or scanning. Modeling these queue dynamics enables the simulation to realistically capture operational bottlenecks and waiting times, which are essential for evaluating KPIs such as truck waiting time, maximum queue length, and overall system throughput.

Utilizing mathematical models and simulations directly addresses **subquestions 5 and 6** head on as the former specifically targets creating a model while the latter focuses on simulating using software.

4
Current state analysis

In this section, the complete overview of the facilities and processes as they currently operate is detailed, beginning with a physical breakdown of the terminals and their respective layouts, machinery, and personnel before moving into a comprehensive breakdown of the present processes, procedures, and operations relating to the night-time delivery process of express shipments. Next, the characteristics of the express packages that are handled in these processes are detailed. The existence and location of the primary bottlenecks and inefficiencies in these processes are then identified. Lastly, the key performance measurements and indicators for how the current process is performing on which the results of the future solution design will be compared to are then presented.

4.1. Current physical breakdown

The physical facilities and equipment are vital to analyze when updating an outdated and inefficient process as these are frequently the source of many bottlenecks that hinder operations. For this project, this analysis will be done on a per-terminal basis. This section will analyze and answer subquestion 1 reiterated below:

SUBQUESTION #1 - TO BE ANALYZED

What are the size, layout, and capacity constraints of the physical spaces for the current processes and the proposed design and what are the quantities, types, physical characteristics, and storage patterns of cargo being delivered?

4.1.1. KLM Cargo terminal layout and external facilities

The current cargo terminal layout at KLM Cargo encompasses three adjacent buildings: **Terminal 1**, **Terminal 2**, and **Terminal 3**, located to the southwest of the main passenger terminal complex and just to the north of the Kaagbaan runway at Amsterdam Schiphol Airport. All the terminals sit adjacent to each other with Cargo Terminals 2 and 3 conjoined. Terminal 1 flanks terminal 2 and is physically connected only by a small, elevated walkway on the first floor.

Figure 4.1 shows where the KLM Cargo terminals are located in the central terminal area of Schiphol. This photo has north pointing upwards.

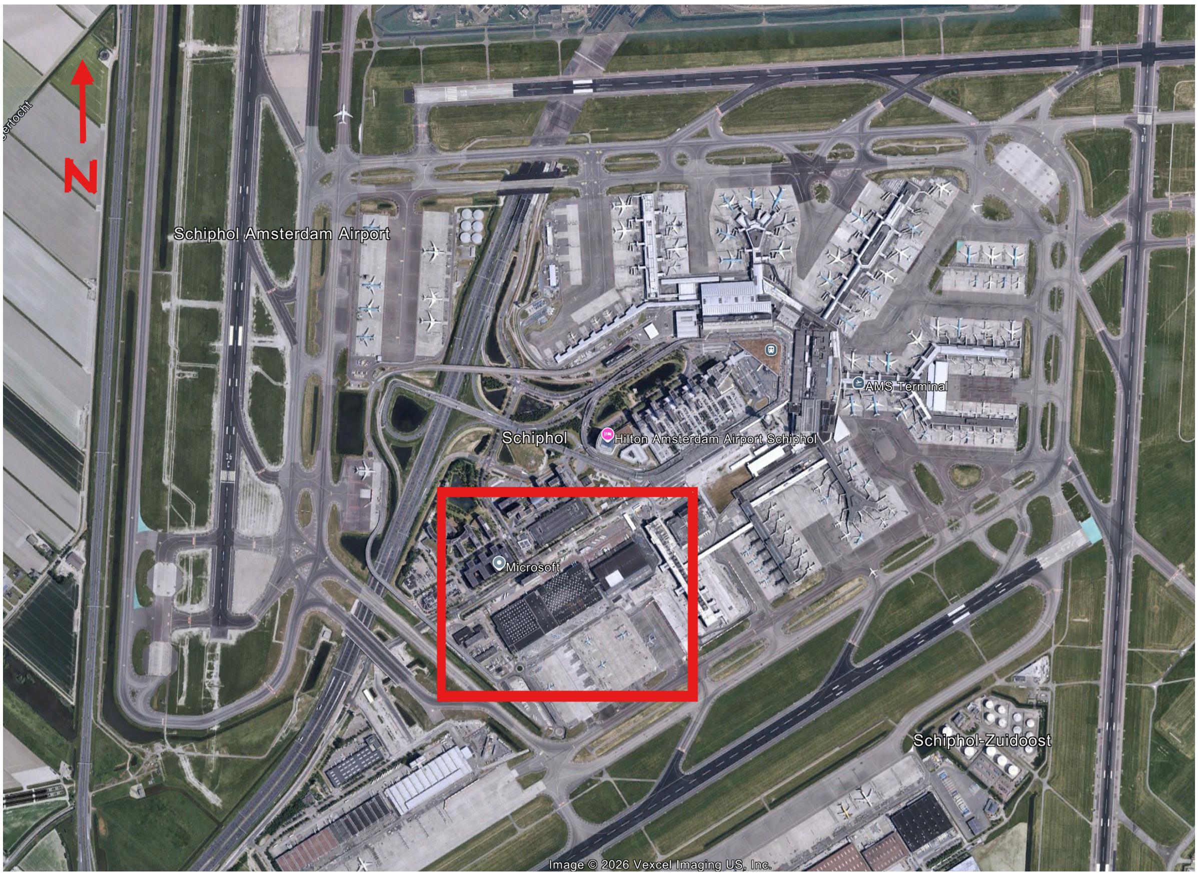


Figure 4.1: KLM Cargo terminal locations at Schiphol *(Source: Google Earth)*

Figure 4.6 is zoomed in aerial image of KLM Cargo's facilities showing the precise size and layout of the terminals and the adjacent external areas.

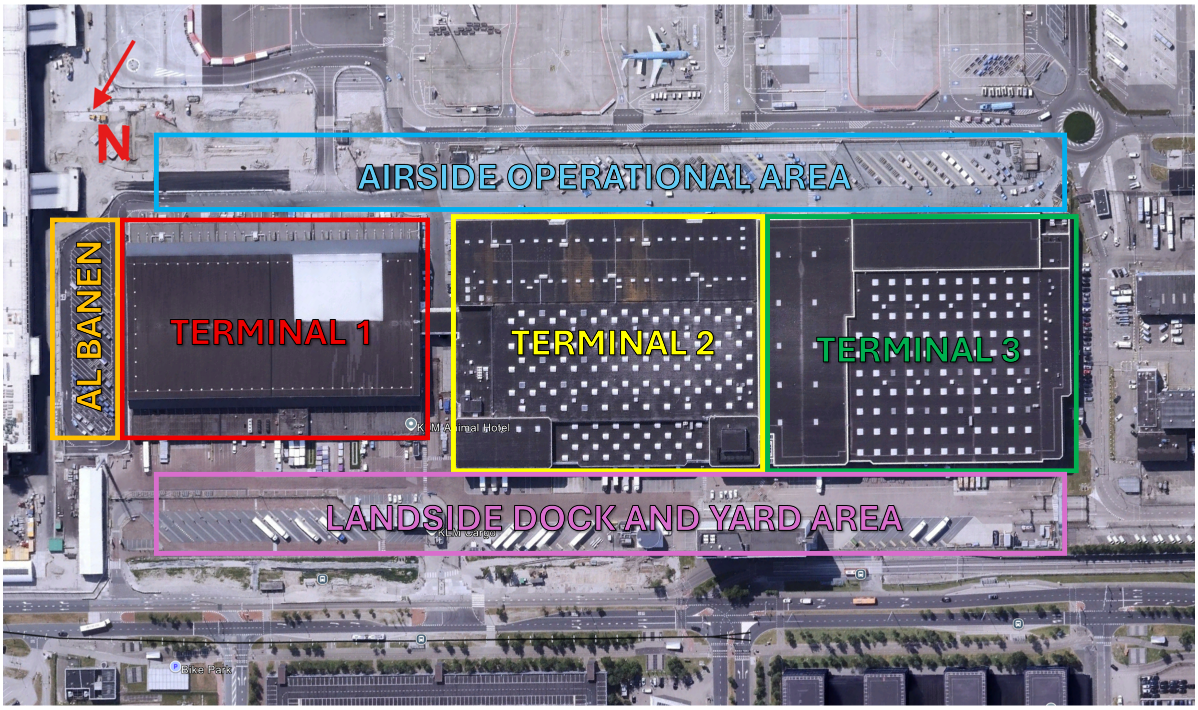


Figure 4.2: KLM Cargo terminals aerial image *(Source: Google Earth)*

From this point forward, all directions, location descriptions, diagrams, and figures will follow the terminal layout shown in Figure 4.6. This orientation is also consistent with the terminal blueprints to maintain clarity and consistency.

Surrounding the terminals are three relevant external areas:

- **Landside dock-and-yard area**
  Immediately to the north of the terminals sits the landside dock-and-yard area where trucks, vans, and lorries arrive and depart at KLM Cargo. It is within these premises that cargo are accepted, checked, docked, unloaded, and loaded.
- **Airside operational area**
  The south-side of the of the terminals encompasses KLM Cargo's airside operational area. This is where export packages are sent out from on their way to aircraft and where import or transito packages (see 4.2.2) arrive from inbound aircraft. This is where the buildup and breakdown area is located which is where bulk shipments are either assembled or dismantled.
- **AL banen**
  Lastly, adjacent to Terminal 1 to the northeast lie the "AL Banen" which are lanes where carts of bulk cargo and ULDs are temporarily stored before being handled.

4.1.2. Terminal 1

Terminal 1 (T1) handles smaller and more specific cargo, such as mail/post, express packages, animals, and valuables. Below in figure **??** is a detailed blueprint schematic of the terminal showcasing the important aspects that relate to the project at hand.

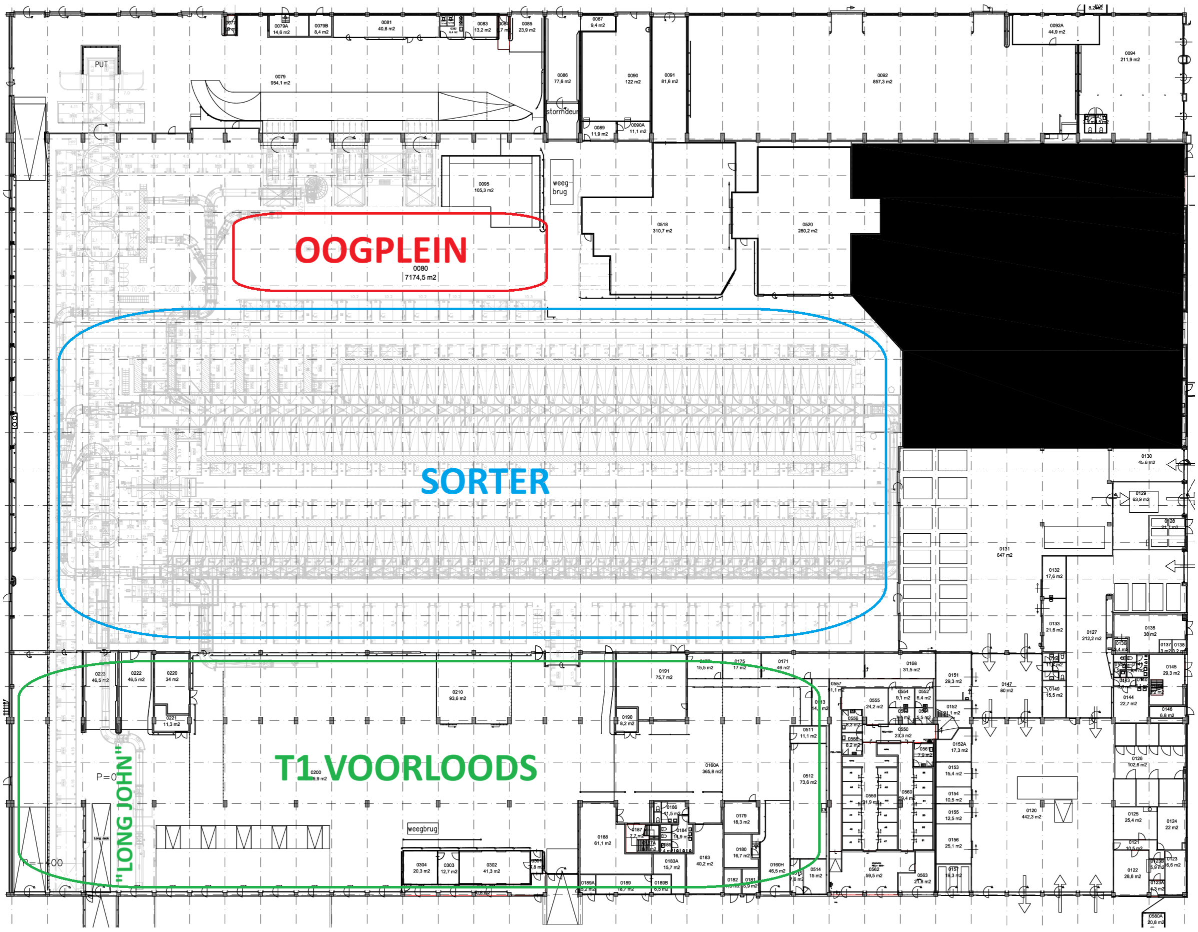


Figure 4.3: Blueprint for KLM Cargo Terminal 1 *(Source: KLM Cargo)*

The primary areas and components of T1 relating to this project are identified below:

- **Sorter**
  Taking up most of the floor space and volume within the terminal is whats called the "sorter" which is a massive automated collection of machinery consisting of conveyors, elevators, carts on rails, chutes, and more. Nearly all shipments in T1 including packages, parcels, bags of mail, etc. are handled in the sorter. Once a shipment is inserted into the sorter at one of the various input locations, it is automatically sent to the proper chute where it will then be subsequently scanned and loaded manually onto an already waiting cart.

  The sorter consists of two major parts: the cart sorting system and the conveyor system

  - **Cart sorting system**
    The cart sorting system comprises the movable lift and rail machinery that physically move carts of packages to and from buffer racking, chute locations, and loading/unloading areas where the carts can be taken from or put on by vehicles heading to or from aircraft
  - **Conveyor system**
    The conveyor system runs separately from the cart sorting system and moves packages and bags of mail around and delivers them from their input points to the proper chutes where they will be subsequently loaded onto the carts by warehouse workers that the sorting system moves around

- **Oogplein**
  The "Oogplein" is the circular area on the ground adjacent to the sorter where the T1 warehouse personnel receive belly carts full of XPS packages and scan them one-by-one into the sorter.

- **T1 voorloods**
  At the bottom left is T1's "voorloods" or acceptance bay where trucks park during the day and unload and load their cargo. In this area, an extension of the conveyor belt system connecting to the sorter called "Long John" comes through.

4.1.3. Terminals 2 and 3

Terminals 2 and 3 work in tandem, with the former handling all imports coming from the airside and the latter the exports coming in from landside. The cargo handled in these two terminals predominantly consists of europallets and unit load devices (ULDs), but also occasionally handles cargo that require special handling such as vats, human remains, perishables, dangerous goods, and more.

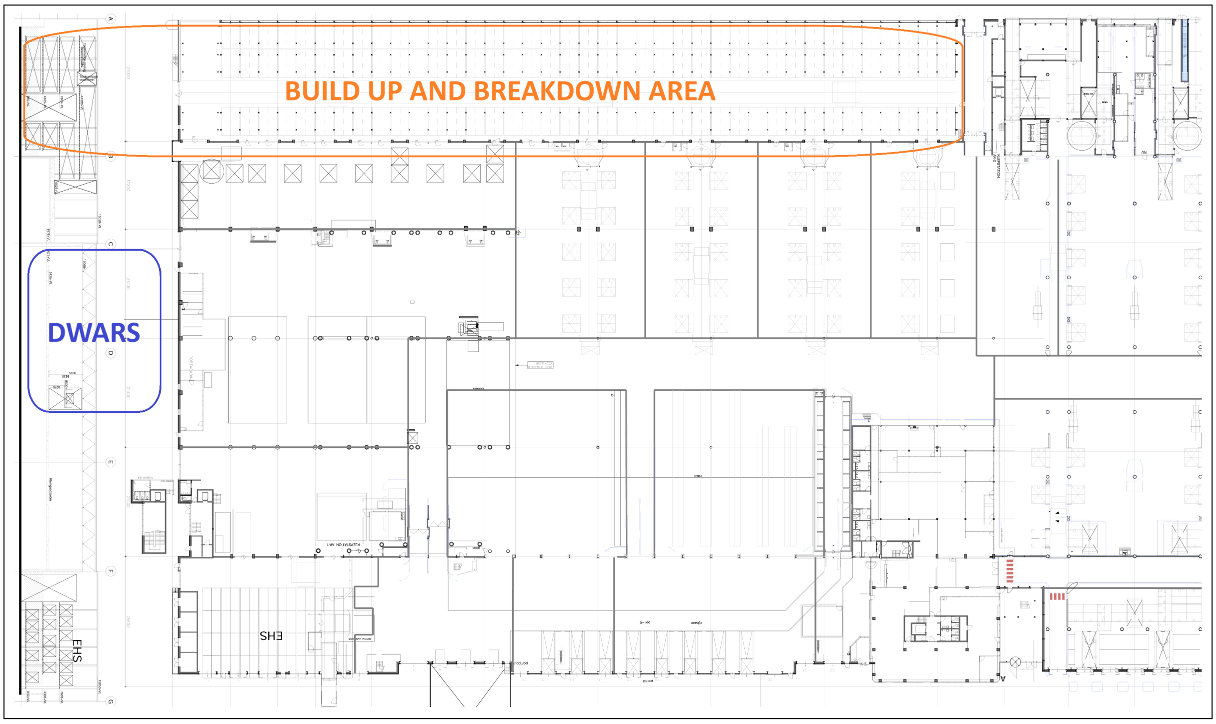


Figure 4.4: Blueprint for KLM Cargo Terminal 2 *(Source: KLM Cargo)*

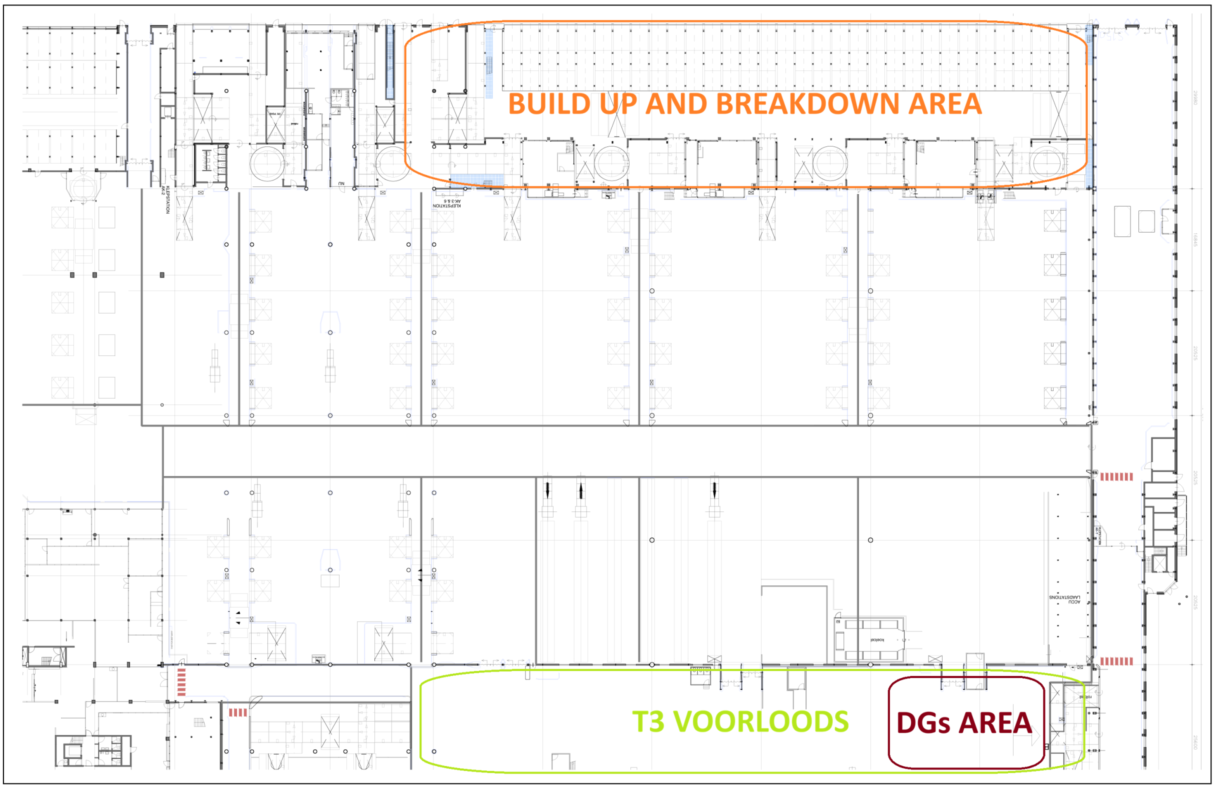


Figure 4.5: Blueprint for KLM Cargo Terminal 3 *(Source: KLM Cargo)*

The primary areas and components of T2 and T3 relating to this project are detailed below:

- **T3 voorloods**
  Similar to that of T1, T3 also has a narrow front entrance bay or "voorloods" where trucks arrived to unload their cargo. This is where the unloading, acceptance, and temporary storage of express packages takes place during night-time operations. The belly carts where XPS packages are temporarily stored are located at the far right side of the T3 voorloods.
- **Dangerous goods area**
  In the voorloods of T3 also sits the Dangerous goods acceptance area where all cargo containing dangerous goods are sent to in order to endure additional and proper checks and procedures.
- **Buildup and breakdown area (B&B)**
  The build up and breakdown area (B&B) is located at the top of T2 and T3 and forms the region between the warehouse and the external airside operational area. It is in this location where bulk shipments, including those with XPS packages, are either assembled or dismantled.
- **DWARS**
  In the far left area of T2 close to the door leading towards T1, there exists an area called "DWARS" which contains belly carts where XPS packages coming from B&B are temporarily stored before being sent to T1.

4.1.4. XPS package types and characteristics

It is vital to get a clear picture of the types of packages that are being handled in the process that KLM Cargo would like to improve as many factors of the design depend on the characteristics of these packages, including storage capacity, required physical space, and dimensions of conveyors and scanning equipment.

- **Special handling codes**
  While most cargo handled is simply labeled "general cargo" and does not require special handling or specific care, that is not the case for all cargo. The International Air Transport Association

(IATA), which is the trade association for the world's airlines, has produced a list of standardized *special handling codes* for cargo that requires additional processing steps or unique methods of handling, whether it be during transportation or storage [11]. These take the form of three letter initialisms. Express shipments are classified "XPS" which KLM Cargo also assigns a product code of M21. These are the primary cargo type that this project focuses on and indicates that this specific shipment is to be handled by the Express and Postal System (EPS) department at KLM Cargo.

Packages and cargo may have more than one special handling code and there are many codes that take precedence over XPS if they are also part of a shipment. For example, should an XPS package also contain dangerous goods (DGD) or human remains (HUM), then these are not handled in T1 but instead are sent to the dangerous goods acecptance area in the voorloods of T3.

- **Physical characteristics**
  Per Dutch regulations as well as listed in the KLM Cargo product portfolio, weights of packages must be below 23kg for humans to lift. Any XPS package that is over this weight is deemed "Express Heavy" and given a KLM Cargo specific code of "XPH". These packages are always sent to T3 to be handled there as they are both too heavy for the warehouse personnel to handle as well as for the sorter. XPS shipments are allowed to comprise multiple packages but each package must remain under 23kg and the entire shipment must be under 300kg total [1].

  Also per the product portfolio [1], the volume of XPS packages cannot exceed 130 x 80 x 80 cm or it will be handled as general cargo in T3 as well. It can also not be too thin on one side as the sorter cannot handle these types of packages as well [D.1.8].

- **Package condition**
  It is important to keep track of the condition of the cargo as damaged cargo or cargo that has been tampered with is not allowed to be accepted. For example, packages showing leakage, damaged containers, packages with multiple kinds of tape, with holes or other strange markings, or presenting a weight or volume that does not match what is listed on the AWB are to be flagged, not accepted by KLM Cargo, and sent back to the customer.

SUBQUESTION #1 - ANSWERED

In summary, this subquestion was answered by analyzing KLM Cargo datasets, reviewing operational documentation, conducting interviews with KLM Cargo personnel, and performing on-site observations of the terminals. These methods allowed the quantities, types, and physical characteristics of the handled cargo to be identified, while the layouts and capacity constraints of the terminal spaces were mapped and documented. Together, this information provides a detailed representation of both the cargo characteristics and the physical environments in which the current and proposed processes operate.

4.2. Current procedural breakdown

This section will detail the procedures in place for the current night-time delivery processes. This allows for the second subquestion to be addressed:

SUBQUESTION #2 - TO BE ANALYZED

What are the sequential steps of and bottlenecks present in the current night-time delivery process, and which KPIs are appropriate to best evaluate and compare both the current and proposed processes?

There are three primary flows within the KLM Cargo terminals for incoming XPS shipments to be handled at the Oogplein in Terminal 1. These comprise **bulk flow**, **transito flow**, and **export flow**.

Figure 4.6 shows the locations for the temporary storage of XPS packages as well as the travel routes the XPS packages take for each flow.

All flows end at the Oogplein but originate in different areas around the terminal complex with bulk flow coming from the AL Banen (location "3" in Figure 4.6), transito flow coming from the DWARS location at the left end of Terminal 2 (location "2" in Figure 4.6), and export flow originating in the Terminal 3 voorloods (location "1" in Figure 4.6).

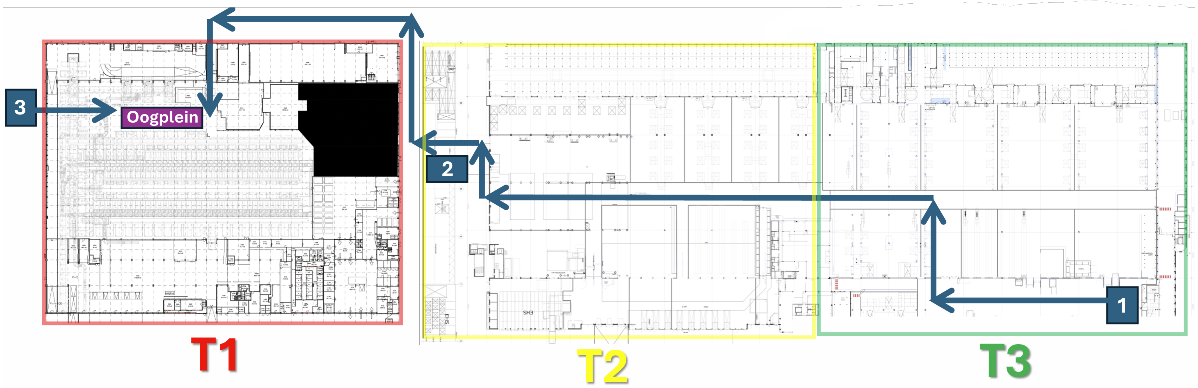


Figure 4.6: Map of XPS flows *(Source of background images: KLM Cargo)*

The main focus for this project will be on the export flow, as this is the flow that causes the most problems during the night and morning handover; however, all three flows impact the performance of the night-time process and all have effects on each other and therefore it is important to understand each flow.

4.2.1. Bulk flow

The first flow type involves XPS packages arriving from flights in XPS-only ULDs or pallets. These containers only consist of XPS packages and therefore do not need to be broken down at B&B as is necessary with most ULDs or built-up pallets which need to first be handled at T2 (see transito flow below). These XPS-only bulk pallets are driven and stored in the AL banen next to T1 and are unloaded and inserted into the sorter in the morning when T1 reopens.

4.2.2. Transito flow

The second flow type of XPS packages comes from pallets coming off airplanes that are a mix of general cargo and XPS. These combined pallets and ULDs are sent to the B&B area outside and within T2 where they are broken down and sorted by package type. All XPS packages that are under 23kg are then loaded onto belly carts at the DWARS location near the end of T2 which are then transported to the Oogplein in T1 in the morning by employees of T1.

4.2.3. Export flow

The export flow is the most important XPS flow relating to this project as it is the process that causes the most inefficiencies KLM Cargo wishes to address. For the export flow, clients deliver cargo and freight to the KLM Cargo facilities on a 24/7 basis. Trucks arrive to the premises with only general cargo, with only XPS, or with a mix of both. During the day, trucks with only general cargo go straight to T3, while trucks containing any amount of XPS packages must first offload those at T1 before traversing either to T3 for further unloading/loading or to the exit if it only had XPS. As T3 is open at all hours of the day, it can receive the trucks as normal. However, with T1 closing at night, night-time deliveries of XPS products destined for T1 must also be handled and temporarily stored in T3 until the former reopens in the morning.

This night-time process is divided into three sub-processes: **standard acceptance**, **non-standard acceptance**, and **morning handover**, each of which is broken down into a step-by-step procedure below. It is important to note that the procedures presented represent the most commonly observed operational sequence. In practice, however, the process is not always strictly standardized, and variations may occur depending on operational conditions, cargo characteristics, staff decisions, and other situational factors.

Standard delivery process
The following is the step-by-step process for a standard acceptance of XPS shipments during the night:

1. Truck arrives at the premises and is checked documentation and receives a token
2. Truck drives to T3 and waits in the queue for an available dock
3. Truck receives word that a dock has become available and it parks at this dock
4. Truck unloaded of all its pallets, regardless of the mix of XPS and general cargo
5. Pallets are placed one-by-one in a line in the voorloods by one forklift driver
6. A second forklift driver at the other end of the line removes the pallets one-by-one and for each one, checks if it is acceptable and if it has any special handling codes, determines it is neither, and brings it to its correct location:
   (a) If general cargo, it is driven to the proper buffer or storage location within the warehouse
   (b) If XPS, the driver exits the forklift, manually scans and checks the package before walking or driving it to the belly carts waiting at the end of the voorloods
7. This is repeated until all pallets have been unloaded from the truck and removed from the line

Non-standard delivery process
The following is the step-by-step process for a non-acceptance of XPS shipments that require special handling codes:

1. Truck arrives at the premises and is checked documentation and receives a token
2. Truck drives to T3 and waits in the queue for an available dock
3. Truck receives word that a dock has become available and it parks at this dock
4. Truck unloaded of all its pallets, regardless of the mix of XPS and general cargo
5. Pallets are placed one-by-one in a line in the voorloods by one forklift driver
6. A second forklift driver at the other end of the line removes the pallets one-by-one and for each one, checks if it is acceptable and if it has any special handling codes, determines it is acceptable but has a SHC, and brings it to its correct location:
   (a) If general cargo, it is driven to the proper buffer or storage location within the warehouse
   (b) If XPS, the driver exits the forklift, manually scans and checks the package before walking or driving it to the proper location within the warehouse (e.g. to the dangerous goods acceptance area or to the build up and breakdown area for XPH packages)
7. This is repeated until all pallets have been unloaded from the truck and removed from the line

Morning handover
The following is the step-by-step process for the morning handover procedure in which temporarily stored XPS packages in T3 are shuttled over to T1 in the morning as determined by interviews and werkinstructies [5] as well as corroborated with in-person observations:

1. T1 reopens at 06:00
2. Warehouse personnel come to T3 voorloods to pick up the belly carts (done by T3 personnel Sunday through Friday and T1 personnel on Saturdays)
3. Belly carts full of export XPS shipments is driven through T3 and T2 over to T1 where they are dropped off
4. T1 warehouse personnel then scan the XPS packages one-by-one until they are all sorted

4.2.4. Comparison with day-time delivery process

During the day, T3 sees no standard XPS export acceptances. All trucks containing only general cargo are sent directly to be T3 while all trucks containing only XPS packages are sent directly to T1. Trucks containing a combination of general cargo and XPS packages are sent to T1 first to have all the XPS offloaded before being sent to T3 to have the general cargo taken off. When a truck arrives at T1, it docks and is offloaded with the XPS packages being checked and, if accepted, sent into the sorter.

4.3. Identified bottlenecks and inefficiencies

The problems listed in section 1.3.2 describe the negative effects experienced in the current night-time delivery process, such as delays in XPS package processing, truck congestion, and T3 operational disruptions, the bottlenecks discussed in this section refer to the underlying causes that produce these problems.

Through interviews with KLM Cargo personnel, analysis of data, review of company documents, and on-site observations, several recurring bottlenecks within the night-time delivery process were identified. These bottlenecks slow down cargo handling operations, increase congestion, and contribute directly to the problems previously described. The most significant bottlenecks are outlined below:

1. **XPS operations take up space in T3 voorloods**
   The temporary handling and storage of XPS packages in T3 takes valuable space within the already relatively small footprint of the T3 voorloods. When removed from trucks, XPS packages are situated on large pallets similar to how general cargo is stored, despite being a fraction of the size and weight. This inefficient use of space reduces the available capacity for normal T3 operations and contributes to congestion during peak periods.

   Additionally, the belly carts that sit in T3 during the night also take up space at the far end of the voorloods near the dangerous goods acceptance area which gets in the way of that department and also reduces the amount of floor area available for T3 operations as well.

2. **XPS and general cargo share equipment and personnel**
   The handling of XPS packages also interferes with T3 operations by utilizing the same equipment and warehouse workers that handle the general cargo. These resources are now no longer only working on their primary objective of general cargo but now need to split their time and energy to handle XPS packages as well. These workers also need to get off their forklifts in order to scan the XPS packages and bring them to the belly carts which wastes additional time that could be used for general cargo operations.

3. **Inefficient and disruptive transport route during the morning handover**
   The transfer of XPS cargo from T3 to T1 when the latter reopens requires a warehouse worker to drive the belly carts through a long route via T3, T2, and outside before finally reaching T1. Not only is the belly cart placement in the voorloods located in the farthest possible region within the terminal complex from the Oogplein, this route takes them through active working zones. This then can interfere with ongoing warehouse operations in these terminals and becomes especially problematic during busy periods, increasing travel times and creating additional operational disruptions. Additionally, the travel time outside can cause damage to packages if the area is experiencing inclement weather.

4. **Overloading of the Oogplein**
   When T1 reopens in the morning, all accumulated XPS packages delivered during the night are transported to the Oogplein area to be processed into the sorter. As large quantities of packages can accumulate in the belly carts, particularly during peak periods, this then overwhelms the workers at the Oogplein as they all arrive at once instead of being processed continuously throughout the night. This leads to delays in sorting and greatly increases the likelihood that XPS packages miss their intended flights.

5. **Organizational and procedural inconsistencies**
   Many delays and lost time occur due to inconsistencies in procedures and a lack of communication and coordination between T1 and T2/T3. Work instructions are not always clearly defined or consistently followed, which can lead to confusion and finger-pointing between warehouse personnel and variations in how the process is executed. A lack of clear guidelines as to who is

responsible for what tasks when cause important procedures such as the morning handover to occur far later than required, with no back up system alerting workers that this has not been done.

6. **Lack of prioritization of express shipments**
While officially the XPS shipments arriving at T3 during the night are meant to be placed on belly carts with respect to their flight departures, this is not done. They are simply placed on any of the belly carts available. While this is not as crucial on light days when only a few packages are accepted, during peak periods this is a major contributing factor to packages missing flights as packages then arrive in an arbitrary order and can lead to time-sensitive XPS packages being inserted after non-urgert shipments.

These bottlenecks collectively create the operational conditions that lead to the problems described earlier, including delayed shipments, missed flights, increased congestion, and disruptions to normal T3 operations. Identifying these root causes is vital for developing an improved delivery process that eliminates or mitigates these inefficiencies.

4.4. KPI-based performance assessment of current process

In order to objectively assess the performance of the current night-time delivery process of XPS shipments and later compare it with the proposed design, the KPIs established in Chapter 3 are applied to the current process. These KPIs provide measurable criteria for evaluating operational performance and identifying areas where improvements can be achieved.

As described previously in section 3.2.6, the KPIs are divided into two categories: **static** and **dynamic**. Static KPIs describe structural characteristics of the system and can be determined directly from the process mapping and facility layouts while dynamic KPIs represent operational performance and require modeling and simulation of the process to obtain values and measurements. At this stage of the analysis, only the static metrics of the current process can be determined directly. Dynamic KPIs will be calculated later through the simulation models described in Chapter 7.

4.4.1. Static process metrics - current process

Below are the static KPIs applied as well as their values for the current process:

- Total distance cargo travels within the system:
  - Bulk flow: 121 m
  - Transito flow: 187 m
  - Export flow: 570 m
- T3 floor area used for temporary storage of XPS packages: 120m^2
- T1 floor area used for current process conveyors ("Long-John"): 75m^2
- T3 floor area used for XPS packages during truck unloading: 0.96m^2 per europallet used (one third of T3 operations at night D.1.7)

These were obtained by examining the blueprints provided by KLM Cargo as well as corroborated with in-person observations of the areas.

4.4.2. Dynamic performance KPIs

The dynamic KPIs defined in Chapter 3 will be used to evaluate the operational performance of both the current and proposed solution processes. These indicators require simulation of the process and will therefore be calculated later using the DES models developed in Chapter 7.

SUBQUESTION #2 - ANSWERED

In conclusion, the comprehensive breakdowns for the three process flows, being bulk, transito, and export, were determined through interviewing KLM Cargo employees, reviewing work procedural documents, and examining the blueprints of the terminals. Additionally, they were corroborated with in-person observations of the processes. Based on this analysis, the key bottlenecks in the system were identified and the relevant KPIs for evaluating both the current and proposed processes were established.

5

Requirements definition

In this section, the requirements and wishes for the project are outlined. First, the relevant stakeholders and their respective relation and objectives are described. Secondly, the objectives that the proposed new design will aim to achieve are listed, followed by a detailed description of what an ideal scenario would look like from start to finish. Next, the limiting factors that will hinder the ability to achieve this ideal future state are listed. Lastly, the potential risks that may come up and the preventative measures instilled in response are provided.

This chapter will effectively answer the third subquestion presented in section 1.4.2, being:

> **SUBQUESTION #3 - TO BE ANALYZED**
>
> What regulatory, safety, security, and stakeholder requirements must the new delivery system meet, and what relevant technologies, systems, or operational practices from other companies or industries could be incorporated into the final design?

5.1. Stakeholder analysis

In this section, the stakeholders for this project are listed and their relevance, influence, and power relating to this project are explored in detail.

5.1.1. Stakeholder determination

The stakeholders for this project were primarily determined through the various interviews with KLM Cargo personnel, who shared the parties that were most involved, most impacted by the inefficiencies of the current process, and most wished to see improvements and change.

5.1.2. Relevant stakeholders

The following stakeholders hold direct or indirect relevance to the night-time XPS delivery process and therefore to this project:

- **Truck drivers**
  Truck drivers delivering XPS packages to the terminals interact directly with the night-time delivery process. They are therefore strongly affected by any changes to the system. While they have little influence over the design of the process itself, they have a high level of interest because they experience its operational drawbacks firsthand, such as long waiting times, package delays, and the risk of missed acceptance deadlines.
- **Trucking companies / agents / customers / clients**
  Trucking companies and clients (such as UPS, DHL, etc.) organize the transport of XPS packages

to the cargo terminals and coordinate deliveries. Although they are not directly involved in the operational handling of shipments inside the terminal, they have a higher level of influence as they can decide which airlines or cargo handlers to work with, meaning that inefficiencies and delays can influence their choice of service provider. As a result, they have a strong interest in a redesigned process that improves reliability, reduces delays, and ensures efficient handling of XPS packages.

- **Shippers (individual consignors)**
  The shippers are the individuals who are sending the XPS packages from one location to another, such as a person sending a package from New York to Warsaw. These shippers pay extra for express shipping and thus do not want to experience any delays or slowdowns in getting their package to its destination. They do not hold much power and do not have as high an interest due to them dealing with the freight forwarders rather than KLM Cargo directly, but continued delays might cause them to switch to other trucking companies,
- **KLM Cargo**
  As the organization responsible for operating the cargo terminals, managing the personnel and various processes, and implementing improvements, KLM Cargo holds the highest level of decision-making authority in implementing new processes. The company ultimately determines operational procedures, infrastructure investments, and strategic direction for cargo operations based on reports, data, and intel from its various departments.
- **Warehouse personnel (Terminal 1 and Terminal 3)**
  Warehouse workers execute the day-to-day operational tasks such as cargo acceptance, scanning, transport, and sorting. They interact directly with the night-time cargo delivery process and therefore have high interest in seeing this process be improved. While they do not hold as high power as managerial positions, they hold high interest in the project as their experience is invaluable and ensuring the new process works well with their objectives is crucial.
- **Unit managers (Terminal 1 and Terminal 3)**
  Unit managers oversee the operations within their respective terminals and supervise warehouse personnel. They hold moderate decision-making authority and play a key role in proposing new systems and ensuring changes are implemented well into their operations.
- **BPI (Business Process Improvement) Department**
  The BPI department is responsible for identifying operational inefficiencies and developing improvement projects within KLM Cargo. They directly investigate and research new processes, systems, and proposals that can bolster KLM Cargo's operations and present their findings with high levels of influence. As the department that initiated this project, it holds significant influence in evaluating proposed solutions and guiding process optimization efforts.
- **EPS (Express and Postal Solutions) Department**
  The EPS department is responsible for handling express shipments within the cargo network and is the operational unit receiving and processing XPS packages. Because the project focuses on improving the express cargo delivery process, this department has a very high level of operational interest and influence.
- **C&S (Compliance and Safety) Department**
  The Compliance and Safety department ensures that all cargo operations adhere to regulatory requirements, security protocols, and safety standards, both internally within the company's own guidelines and externally with official regulations and governmental legislation. They hold significant authority in determining whether proposed operational changes meet regulatory and safety requirements for any project, but hold less interest in this specific project.
- **S&D (Sales and Distribution) Department**
  This department represents the commercial side of KLM Cargo and is responsible for promoting and distributing cargo services to customers. While they are not directly involved in the operational handling of XPS packages, improvements to the night-time delivery process of these shipments can influence product offerings, service reliability, and market competitiveness. Their role is mainly to ensure that operational changes align with customer expectations and the service promises made to clients.

- **Product engineers and product owners (XPS products)**
  These stakeholders are responsible for designing and managing the express cargo product within KLM Cargo's portfolio. They influence how operational capabilities align with product specifications and service levels.
- **iCargo/IT team**
  The iCargo team manages the digital cargo management system used to track shipments and coordinate operations. Because this system plays a central role in the operations of the cargo handling processes and also in recording and collecting data to be used by other departments, it is vital that the new process be fully integrated with this software. This team therefore has considerable influence on how new operational procedures can be integrated digitally.
- **Equipment suppliers (e.g. Lödige and Vanderlande)**
  These external suppliers provide key infrastructure such as sorting systems and conveyor equipment used in the cargo terminals. While they do not control operational decisions, their technologies influence what technical solutions are feasible within the system.

5.1.3. Stakeholder power grid

The overall power and influence values for stakeholders were assessed based on their overall relevance and relation to the project and its outcome. These were then balanced and placed on a grid relative to each other as shown below:

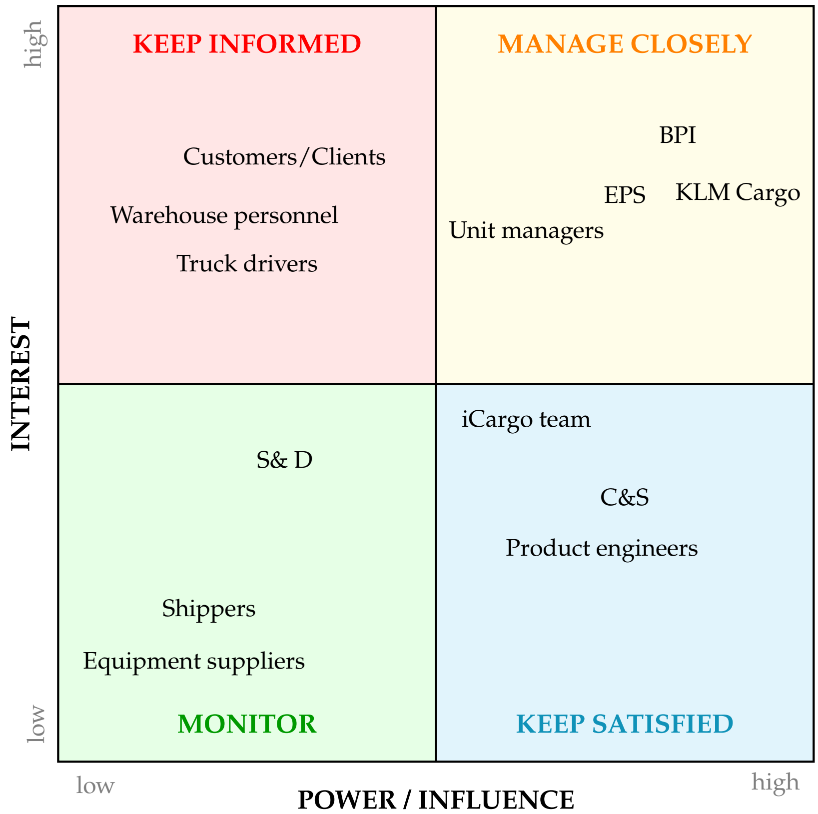


Figure 5.1: Stakeholder power–interest grid for the night-time XPS delivery process

5.2. Desired objectives and criteria

Building off of the general and project-specific goals listed in section 1.3.3, a list of specific objectives and criteria is established which will define how the final, optimized design will achieve those goals. These are as follows:

1. **Enable direct drop-off of XPS shipments at Terminal 1**
   Allow XPS to be delivered directly to T1 during night-time operations, reducing the amount of

trucks going to T3 and contributing to its congestion and workload.

2. **Improve operational efficiency and cargo flow**
   Streamline the acceptance and handling process to reduce processing time, eliminate unnecessary transport between terminals, and minimize congestion and space usage in areas such as the Oogplein and the T3 voorloods.
3. **Improve prioritization and delivery reliability**
   Ensure XPS are handled and sorted by priority so that time-critical shipments are handled first and less packages miss their intended flights.
4. **Enable automated and retain 24/7 continuous operations**
   Develop a new process capable of autonomous XPS acceptance both physically and procedurally, including automated scanning, weighing, validation against shipment data (FWB), and verification of acceptance criteria, minimizing human intervention.
5. **Ensure regulatory compliance and operational safety**
   Maintain full compliance with customs regulations, security requirements, and "Ready for Carriage" standards, including the ability to detect damage, leaks, dangerous goods, and other special handling conditions.
6. **Integrate seamlessly with existing systems and infrastructure**
   Ensure compatibility with the current sorter system, digital cargo management systems (e.g., iCargo), and existing operational procedures to allow smooth system integration.
7. **Provide clear procedures and user-friendly system interaction**
   Design the system to be intuitive and user-friendly for drivers and warehouse workers, including clear acceptance criteria, standardized procedures, and accessible support in case of issues or questions.
8. **Ensure operational robustness and fallback procedures**
   Include backup procedures, exception handling, and contingency plans to ensure operations can continue in case of system failures or non-compliant shipments.
9. **Allow flexibility and scalability for future operations**
   Ensure the system can handle varying cargo volumes, accommodate special cargo types, and potentially support similar automated operations during daytime periods.

5.3. Ideal scenario

An ideal design for a new night-time cargo delivery system would be of the following. Trucks containing XPS packages would proceed directly to T1 after leaving documentation. The driver would be able to offload all the XPS packages on the truck and insert them into an available automated, user-friendly, and integrated machine one-by-one. This machine should be able to automatically scan the label and pull up the information of the shipment and confirm everything is correct, measure its weight and volume, detect if the shipment is in good condition and as not been tampered with, and clear it for dangerous contents. It should be able to then make a decision whether to accept the package, flag it for additional human checks, or reject it outright and send it to T3 for human checks. Ideally, all general XPS shipments should be able to use this system with the only packages to be sent to T3 being the ones with special handling codes or damaged/potentially tampered with. Operations at T3 will now no longer contain general XPS shipments and can continue essentially unaffected with the space previously utilized for temporary storage now usable for T3 operations.

5.4. Limiting factors and known constraints

It is important to understand the physical and systematic limitations that exist which present obstacles and hindrances to the realization of any new design for a new night-time cargo delivery system and procedure. The primary limiting factors are listed below with their respective impacts provided in depth:

5.4.1. Physical space

The available floor area and surrounding infrastructure may restrict the size, layout, and capacity of any proposed solution. The new design must be able to fit within and integrate with the existing operations, flows, and physical structures.

5.4.2. Costs

Both initial investment (construction, equipment, materials, etc.) and operational costs (labor, maintenance, energy usage, etc.) must remain within the budget of KLM Cargo.

5.4.3. Hazardous cargo storage

Certain types of cargo require special handling and designated storage areas that comply with safety regulations. A new storage system or delivery procedure must be able to handle these types of cargo or have plans in place for when these types of cargo and packages are delivered.

5.4.4. Weather exposure

Outdoor delivery or storage areas may be affected by rain, wind, or extreme temperatures, which can hinder or delay operations, damage cargo, and degrade the delivery system overtime. Therefore, it must be able to withstand these conditions both for its own longevity and for the conditions of the cargo it stores.

5.4.5. Night-time security measures

Adequate security protocols are needed to prevent theft, tampering, or unauthorized access during overnight operations. Due to limited personnel and staff during the night, the new delivery system must adhere to strict security measures.

5.4.6. Compliance, safety, and security checks

KLM Cargo maintains a Cargo Handling Manual (CHM) [14], [13], [15] that specifies the checks required for shipments to be accepted as *Ready for Carriage* or able to be shipped. Based on the procedures outlined in this manual, several verification steps must be performed before cargo can enter the operational process. The verification checks applicable to all cargo are as presented in the CHM are listed below:

1. **Security clearance verification**
   (a) Verify whether shipments are security cleared
   (b) Visually check that the package is suitably packed to protect its contents
   (c) Visually check that the package is effectively protected from unauthorized interference
   (d) If the package is secured by a regulated agent or known consignor:
      i. Request verification of identification (passport or driver's license)
   (e) If the package is unsecured:
      i. The shipment must be screened before being loaded onto an aircraft
2. **General Ready-for-Carriage checks**
   (a) Compare the FWB with the physical shipment and register any discrepancies:
      i. Airport of departure and airport of destination
      ii. Weight values
      iii. Number of pieces
      iv. Shipper and consignee details
      v. Commodity description ("nature and quantity of goods")
      vi. Physical appearance compared with the shipment description
      vii. Ensure that no undeclared Dangerous Goods (DG) are present
      viii. If DGs are declared, refer to the IATA Dangerous Goods Regulations

   (b) Perform a visual examination of the packaging
   (c) Perform a visual examination of marks and labels
   (d) Ensure compliance with applicable embargoes and operational restrictions
   (e) Verify wooden packaging where applicable
3. **XPS-specific Ready-for-Carriage checks**
   (a) Maximum weight per parcel: 23 kg
   (b) Maximum weight per shipment: 300 kg
   (c) Maximum dimensions: $130 \times 80 \times 80$ cm
   (d) Dangerous Goods are not accepted except DGR/AOG shipments

It is vital that any new system be able to adhere to these checks and fully be able to ensure that these checks are completed.

5.5. Risk management

Effective risk management is critical to ensure that the final design for the nightly cargo delivery process is not only efficient and functional but also safe, feasible, and reliable when uncertain and unexpected events occur. By identifying potential risks and hazards ahead of time and planning ahead by taking these uncertain events into account in the design, the project can prevent future problems, avoid costly mistakes, and ensure that the implemented solution meets all operational and stakeholder requirements.

Key risk considerations include:

- **Safety**
  The safety of the new design must be ensured to protect all individuals who interact with the system on a regular basis. This includes warehouse personnel, delivery drivers, maintenance crews, and any other staff involved in the nightly operations. Risks such as collisions, injury risks, and exposure to hazardous cargo must be identified and mitigated to prevent accidents, injuries, or health hazards.
- **Cost overruns**
  Financial feasibility is a critical risk factor. The final design should avoid excessive costs, whether in terms of capital investment, operational expenses, or long-term maintenance. Without proper consideration, a technically efficient design could become a financial burden, reducing its overall value and viability for KLM Cargo.
- **Security**
  The security of the cargo is crucial. As KLM Cargo handles packages entrusted by external parties, any theft, damage, or loss can severely impact the company's reputation, client trust, and operational credibility. The design must incorporate robust security measures, including controlled access, surveillance, and procedural safeguards, to ensure the safe handling and storage of cargo at all times.
- **Space constraints**
  The physical limitations of the terminal and surrounding areas represent another critical risk. Even the most efficient design is ineffective if it cannot be implemented within the available floor space and operational zones. The new system must fit within the spatial boundaries of Terminal 1 and the relevant areas of Terminal 3, while still allowing for smooth workflow, vehicle access, and future scalability.

SUBQUESTION #3 - ANSWERED

In conclusion, the key stakeholders involved in the night-time XPS delivery process were identified and their respective roles, interests, and levels of influence were analyzed. This was accomplished primarily through interviews with KLM Cargo personnel, supplemented by operational documentation and process observations. From these sources, a comprehensive list of stakeholder objectives, requirements, and operational constraints was compiled. These insights provide essential guidance for the development of the proposed design by ensuring that the solution addresses operational needs, safety and compliance requirements, and the expectations of both internal and external stakeholders.

6
Solution design

In this chapter, the proposed solution design is presented. Taking into account the research, information, and requirements from the previous chapters, this proposal aims to address the identified operational bottlenecks, mitigate or eliminate the problems experienced, and satisfy the requirements and objectives established from the stakeholders.

The chapter begins with an overview of the design development process before describing both the physical and procedural aspects of the new system in detail. Thereafter, the compliance and safety considerations are addressed. Lastly, the expected improvements the proposed solution design will bring are listed.

The fourth subquestion will be addressed in this chapter:

SUBQUESTION #4 - TO BE ANALYZED

How can the nightly cargo delivery process optimally be designed with the information obtained through the previous sub-questions taken into account in order to achieve the highest possible improvements?

6.1. Design development

This section describes how the final solution concept was developed, including its origin, the inputs and information used in its development, and the alternatives that were considered but ultimately not chosen and why.

6.1.1. Concept origin

The concept of an automated system for XPS package drop-off was inspired by a KLM Cargo staff member's experience returning beer crates to the local supermarket which involved inserting the crate into an automated drop-off system. The employee noted the simplicity and efficiency of the process and believed that such a system could also be applied to the XPS night-time delivery system as a means to improve the current process and resolve or mitigate many of the inefficiencies that came with it. The idea was presented to the BPI department who ultimately turned it into a Master's thesis project.

6.1.2. Rejected alternatives

During the development of the new process, a few alternative solutions were considered but ultimately rejected. The first and most obvious solution to this problem would be to keep T1 staffed and fully operational at night so that it could still receive trucks 24/7. This had actually been the case in the past, but ultimately the decision was made to close it at night due to high operational costs and not enough business at night to justify keeping it open which is also the reason it was not considered for this project.

Another option was to have warehouse workers be situated at T1 to manually receive the shipments and store them there in belly carts until the morning when they'd be brought to the Oogplein. While this would help alleviate congestion at T3 and also minimize the operational disruptions that the XPS packages bring at night to T3, it still would not address the problem that the Oogplein becomes overwhelmed in the morning and many XPS packages would still miss their flights due to being inserted into the sorter late. Additionally, this would require more staff to work during night shifts which the company aims to avoid due to health, safety, and cost reasons.

6.2. Physical breakdown of solution design

In this section, the physical aspects of the proposed solution system are fully detailed.

6.2.1. System overview

Presented below is a full diagram and a concise description of the physical components of the proposed system from start to finish.

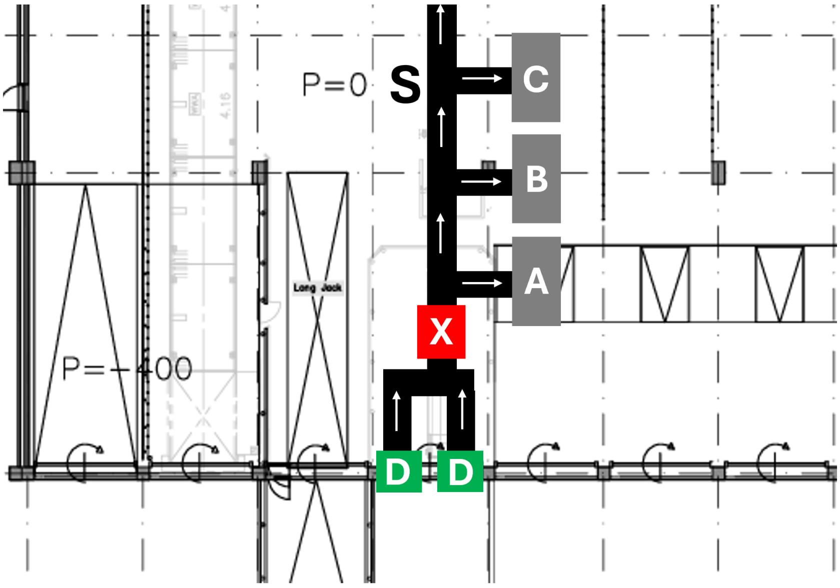


Figure 6.1: New process diagram: D = Drop off machines and user interface | X = X-ray scanner | S = Conveyor and redirecting system | A/B/C = Temporary holding bins *(Source of background image: KLM Cargo)*

1. **Drop-off machines**
   Two self-service drop-off machines, similar to those used in baggage drop-off at passenger terminal check-in areas, are built into the side of T1, in the lower left corner in the area just below where the preexisting sorter connection "Long John" currently sits. As the conveyor system for the sorter comes from Vandelande, it is recommended to utilize them for their baggage drop-off products to help with streamlining construction and costs.
2. **User interface**
   The user interface of the machines will consist of large computer screens with clear and simple instructions and will be able to be changed into multiple languages. Should the driver have an issue or require assistance, a help line and call button to a KLM worker at documentation will be available.
3. **X-ray scanner**
   The single X-ray scanner with built-in AI technology will allow for packages to undergo additional screening even though this is not standard practice currently with KLM Cargo not having X-ray technology on its premises. This will add on security and safety measurements to help offset the removal of human workers. May not be necessary depending on KLM Cargo's security requirements and the fact that packages arrive at the facilities already secure.
4. **Conveyor system**

A conveyor system will connect all these individual parts together. This can consist of either rollers or bands depending on the company contracted. Conveyors will bring packages from the automated drop-off machines to the X-ray machine and also from the X-ray machine to the sorter. As the conveyor system for the sorter comes from Vandelande, it is recommended to utilize them for these conveyors as well so that all systems will integrate nicely.

5. **Redirecting system**
   Redirecting arms or rollers remove XPS shipments that require additional checks or are entered too early from the conveyor and send them into temporary holding bins.
6. **Temporary holding bins**
   Holding bins will be available for temporary storage of XPS packages that are not yet suitable to be inserted into the sorter. Packages that the system identifies as requiring additional human inspection, or those delivered too early for which no sorter chute is yet available, are temporarily stored here until they can be processed by warehouse personnel.

6.2.2. Integration with and modification of existing infrastructure

Firstly, depending on the company and brand of drop-off machine chosen, an additional covering and enclosure outside may need to be constructed to account for the area of the machine to protect it, the drivers, and the packages from weather and outside conditions.

As mentioned in 4.1.2, a preexisting extension of the sorter conveyor system informally called "Long John" extends into the voorloods of T1. The proposed solution system will replace most of its footprint and sit in its place. It is vital that the new system still has a drop off entry point for normal T1 procedures and can perform all of Long John's current uses. The proposed system is expected to take up more floor space than Long John and thus some operations will need to be moved to other areas of the T1 voorloods.

The current conveyor system of the sorter will need to be modified. Presently, the sorter is fully inactive during the night and running the entire system will be too costly as mentioned in 6.1.2. However, the proposed solution design does not require the operation of all machinery, but simply the conveyor system of the sorter which is relatively inexpensive and reliable to operate as opposed to the massive cart sorting system. This system can send inserted XPS packages to the proper dedicated chutes, allowing for them to already be sorted and ready to go in the morning.

Additionally, added sensors would need to be installed along the conveyors to enable automated start/stop functionality as without them, the conveyors would need to run continuously which would waste money and also reduce the durability of the system and potentially lead to breakdowns.

6.2.3. Static process metrics - new process

Below are the static KPIs applied as well as their values for the new process:

- Total distance cargo travels within the system: 10 m
- T3 floor area used for temporary storage of XPS packages: 0 m^2
- T1 floor area used for new process technology and storage bins: 140m^2
- T3 floor area used for XPS packages not accepted by new process during truck unloading: 0.96m^2 per europallet used

These were obtained by examining the blueprints provided by KLM Cargo and overlaying the new design on top of them as shown in Figure 6.1 as well as corroborated with in-person observations of the areas.

6.3. Procedural breakdown of solution design

The new system will consist of four primary procedures:

1. Standard drop-off
2. Non-standard drop-off
   (a) Requires additional human checks

   (b) Inserted too early

3. Rejected drop-off.

6.3.1. Standard drop-off procedure

This is the most commonly expected procedure for XPS package drop-off and will take place when a general XPS package has been inserted into the system that satisfies all acceptance criteria and requires no additional checks.

1. Truck arrives at the premises and is checked documentation and receives a token
2. Truck drives to T1 and waits in the queue for an available drop-off machine
3. Truck parks at drop-off machine when it becomes available
4. Driver vacates truck and uses computer interface of drop-off machine for documentation check, token scan, and process initiation
5. Driver then takes packages one-by-one and inserts them into the drop-off machine
6. Machine closes door and does its checks
   (a) Scans package label and accesses FWB
   (b) Ensures package has no special handling codes that require it to go to T3
   (c) Performs weight and volume checks, ensuring they're within acceptable limits and corroborates the records on FWB
   (d) Scans for damage, leakage, potential tampering, and other flagged conditions using AI technology
7. Package sent to X-ray machine to have its contents scanned
8. Package deemed acceptable with no flags raised sent directly into the sorter
9. The input of XPS packages is repeated until all have been unloaded from the truck and inserted (excluding rejected packages)
10. Driver ends process on the user interface and drivers away

6.3.2. Non-standard drop-off procedure

This is the procedure for two types of non-standard XPS acceptance scenarios:

1. XPS packages that the system flags as "yellow" (see 6.4) due to potential damage, tampering, or unannounced dangerous goods and will require additional human checks after T1 reopens
2. XPS packages that are delivered within two days of a flight departure but for which a chute in the sorter has not yet been assigned and become available

This process will be as follows:

1. Truck arrives at the premises and is checked documentation and receives a token
2. Truck drives to T1 and waits in the queue for an available drop-off machine
3. Truck parks at drop-off machine when it becomes available
4. Driver vacates truck and uses computer interface of drop-off machine for documentation check, token scan, and process initiation
5. Driver then takes packages one-by-one and inserts them into the drop-off machine
6. Machine closes door and does its checks
   (a) Scans package label and accesses FWB
   (b) Ensures package has no special handling codes that require it to go to T3
   (c) Performs weight and volume checks, ensuring they're within acceptable limits and corroborates the records on FWB

(d) Scans for damage, leakage, potential tampering, and other flagged conditions using AI technology

7. Package sent to X-ray machine to have its contents scanned
8. System either deems the package a yellow flag or recognizes that there is no chute yet available and sends the package to the appropriate holding bin for later handling by warehouse personnel
9. The input of XPS packages is repeated until all have been unloaded from the truck and inserted (excluding rejected packages)
10. Driver ends process on the user interface and drivers away

6.3.3. Rejected drop-off procedure

This is the procedure for XPS packages that the system cannot accept and therefore must be brought to T3 for further handling. This includes XPS packages that:

- do not have proper IATA labels
- have special handling codes such as dangerous goods, human remains, cold storage, etc. that require unique handling procedures
- the system flags as "red" (see 6.4)
- that are too heavy (over 23kg), too physically large (larger than 130 x 80 x 80 cm), or too physically thin (less than 2cm on one side)
- have discrepancies between the detected measurements and values listed on FWB

This process will be as follows:

1. Truck arrives at the premises and is checked documentation and receives a token
2. Truck drives to T1 and waits in the queue for an available drop-off machine
3. Truck parks at drop-off machine when it becomes available
4. Driver vacates truck and uses computer interface of drop-off machine for documentation check, token scan, and process initiation
5. Driver then takes packages one-by-one and inserts them into the drop-off machine
6. Machine closes door and does its checks
    (a) Scans package label and accesses FWB
    (b) Ensures package has no special handling codes that require it to go to T3
    (c) Performs weight and volume checks, ensuring they're within acceptable limits and corroborates the records on FWB
    (d) Scans for damage, leakage, potential tampering, and other flagged conditions using AI technology
7. If drop-off machine detects an unacceptable anomaly, the package will be rejected and sent back to the driver with a notification that it will need to be brought to T3 with the token being adjusted
8. If no flag raised by this point, package sent to X-ray machine to have its contents scanned
9. If the X-ray scanner raises a red flag, then the package will be rejected and sent back to the driver with a notification that it will need to be brought to T3 with the token being adjusted
10. The input of XPS packages is repeated until all have been unloaded from the truck and inserted (excluding rejected packages)
11. Driver ends process on the user interface and drivers away to T3 where it will be handled in the same method that is used in the current process with temporary belly carts being driven to T1 in the morning

6.3.4. Morning handover

When T1 reopens in the morning this will trigger two handover processes. The first will be nearly identical to the current process in which belly carts contining all packages that were flagged and sent to T3 for further check that were deemed acceptable in the end by the warehouse workers there will be driven over to the Oogplein in T1 in the morning to be manually sorted into the sorter.

The second morning handover process will require T1 warehouse workers to manually assess and check the yellow flagged packages in the holding bin that the system flagged as yellow and requires additional human checks. They will determine if the packages are acceptable and if they are, directly scan them into the sorter through the new design.

6.3.5. Fall back procedures

Should the system be inoperable for any reason, such as undergoing maintenance or experiencing technical problems, the procedures of the current process will go back into affect until the system can be put back online. It is recommended to write more clear and concise written procedures in the manuals outlying the responsibilities between T1 and T2/T3 workers to address inconsitencies and eliminate confusion.

6.3.6. Night-time vs. day time differences

This process is specifically designed for the night-time operations of the terminals when T1 is closed. The day operations will remain unchanged. Further research and studies can be performed as to whether all XPS deliveries during the day can be handled this way as well as opposed to the current method of having T1 workers unload and handle XPS packages manually.

6.4. Compliance, safety, and security assurance

With the fully automated nature of the new design, it is vital that all safety, security, and compliance checks are performed at or beyond the levels of the current process. Currently, human workers are responsible for checking the condition and documentation of the XPS packages that come through and ensure they do not exhibit suspicious or dangerous qualities and that the package's conditions (weight, volume, package count, etc.) match with what is listed on the AWB. Additionally, warehouse workers are responsible for double checking the drivers documents are correct as well as ensuring the seal on the back of the truck has not been opened or tampered with.

Section 5.4.6 in Chapter 5 lists all the verification checks that must be done before a shipment can be declared secure and *Ready for Carriage*. It is important that the new process be able to accurately, reliably, and consistently be able to perform these checks. This will be achieved through several technologies designed to replicate and enhance the current manual verification checks. Through the combination of IATA label scanning systems, AI visual scanning, automated weight and volume detection, and security X-ray screening technologies, the new process will be capable of performing the necessary checks before allowing packages to be inserted into the sorter.

The following components of the system ensure that the required compliance, safety, and security checks are satisfied:

- **Driver identification and access control**
  - ID scanner and integrated camera verify the identity of the delivery driver
  - Computer checks credentials with the shipment booking and delivery authorization
  - Camera verification provides visual confirmation of the driver and truck in order to maintain security and traceability, allowing for confirmation that the seal was properly secured and removed
- **Automated shipment verification at the drop-off machine**
  - Optical Character Recognition (OCR) and barcode scanners read shipment labels and compare the information with the corresponding electronic documentation (FWB) and checks shipment details such as origin, destination, number of pieces
  - Integrated scale weighs the packages and confirms they comply with XPS requirements

  - Automated dimension scanning confirms that the XPS size does not exceed the allowable limits
- **Visual inspection through computer vision**
  - Multi-range RGB cameras capture high-resolution images of the shipment from multiple angles with AI algorithms, checking for damage, leaks, potential tampering, presence of proper labels and symbols, and that labels are readable
- **Automated security screening**
  - Integrated X-ray screening system inspects the internal contents of the XPS packages to ensure that no prohibited or undeclared items are present
  - Screening results are categorized automatically:
    - **Green flag**: Shipment passes all security checks and is accepted and sent directly to the sorter
    - **Yellow flag**: Shipment is flagged for additional inspection and is diverted to a designated storage bin for additional human checks during daytime operations
    - **Red flag**: Shipment is immediately rejected and returned to the driver due to security concerns or unacceptable contents and must be brought to T3 to be handled by human workers
- **Integration with existing tracking system**
  - Accepted shipments are automatically registered in the iCargo handling system and existing IT systems

Through this combination of automated identification, measurement, visual inspection, documentation verification, and security screening, the proposed system is capable of performing the required compliance and safety checks while maintaining the security standards required for air cargo operations.

6.5. Expected improvements and KPIs

This new system and its corresponding procedures are expected to help in the following ways:

- Reduce the amount of XPS packages that T3 needs to handle during the night
- Reduce the amount of XPS packages that need to be sorted at the Oogplein in the morning
- Reduce the amount of trucks that need to go to T3 during the night
- Reduce the average waiting time of trucks at T3 during the night
- Reduce the average duration that XPS packages take to get inserted into the sorter

Overall, these benefits are expected to greatly improve the operations and reliability of XPS acceptance and handling during the night at KLM Cargo. See Chapter 10 for a more detailed analysis of the overall improvements this system can produce based off of modeling and discrete event simulation.

SUBQUESTION #4 - ANSWERED

In summary, through the information and data obtained via the methods established in Chapter 3, a comprehensive new design could be designed that satisfies all the objectives and criteria laid out by the stakeholders in Chapter 5. This design comprises drop-off machines with built in weighing and dimension scanning technology, an X-ray machine, conveyor belts that connect everything to the sorter, and temporary storage bins for XPS packages that are too early or require additional checks with automated direction changing features that guide packages to the right destination. Additionally, new procedures are put in place.

7

DES Model and Simulation

In this chapter, the Discrete Event Simulation (DES) model and simulation used for this project are presented in detail. First, the relevant data that has been refined and subsequently used in the project are presented along with assumptions made during the refining process. Next, the models used in this project are explained, clearly outlining how the real world processes were transformed into DES components. Thereafter, an overview of the coding and simulation processes that were performed are provided. The chapter ends by discussing how the results were validated, ensuring they match up properly with their real world counterparts.

7.1. Relevant data

This section presents the refined data and charts used in this project. The data was derived from raw operational data provided by KLM Cargo covering June 2, 2025 (Week 23) to January 11, 2026 (Week 2). The analyzed datasets include:

- XPS package data
  - Number of XPS packages accepted per night shift
  - Weight distribution of XPS packages
  - Volume distribution of XPS packages
- Truck data
  - Number of XPS-carrying trucks handled per night shift
  - Distribution of XPS packages per truck
  - Interarrival time distribution of trucks
  - Waiting time distribution of trucks

These datasets are used to parameterize the Poisson distributions applied in the simulation model described in Section 7.2, and to verify and validate the simulation results presented in Section 7.4.

All these data sets can be found in Appendix E.

7.1.1. Assumptions and data cleaning

The following assumptions and data cleaning steps were applied when preparing and analyzing the datasets. These steps were necessary to remove invalid records, exclude outliers, and address missing information:

- An *operational day* is defined as the period from 22:00 to 06:00 (e.g., a package delivered at 02:27 on July 25 is considered part of the July 24 operational day)
- Trucks/packages arriving on weekends are excluded due to their significantly lower frequency

- Records with zero or negative weight or volume values were removed
- Packages with weights above 23 kg or volumes exceeding 0.832 m^3 (larger than $130 \times 80 \times 80$ cm^3) were excluded as they fall outside the operational limits of the process
- All packages with volumes below 0.832 m^3 are assumed to have dimensions within the $130 \times 80 \times 80$ cm constraint
- For multi-package shipments, the total shipment weight and volume are assumed to be evenly distributed across the packages
- In the very few cases when the time that a truck arrived at the counter was not provided, the assigned to dock time or called to dock times were used

7.2. Modeling approach

To effectively compare the current approach to a proposed new design, Discrete Event Simulation (DES) and modeling were utilized. This section will address why this was chosen for this project, list the assumptions made in order to convert the real life scenarios into models, and provide fully detailed descriptions of the models for both the current system as well as the newly proposed system. This section will address subquestion 5 shown below:

SUBQUESTION #5 - TO BE ANALYZED

How can the proposed new design for the nightly cargo delivery process be best represented theoretically as a model?

7.2.1. Model selection

Discrete Event Simulation (DES) was chosen for this project as it fits the time-dependent nature of package delivery. The night-time delivery process of XPS packages consists of sequential processing steps and involves multiple queues, buffers, and routing decisions. Events such as truck arrivals, documentation checks, package unloading, and insertion into the sorter occur at discrete moments in time and change the state of the system which DES is able to model.

Additionally, being able to model both the current and the new processes in this manner allows for direct KPI comparison between the two, which allows for conclusions to be drawn that help support the recommendations that will be put forth at the end of the report.

7.2.2. Model assumptions

It is incredibly complex and time consuming to model real world scenarios 100% due to the limitations of computer software as well as the lack of necessary data required to understand all aspects of a system or scenario. Therefore, it is necessary to make assumptions to simplify the system so that useful results can still be obtained. It is important to simplify the system enough so that the simulation can run yet not simplify it too much that it no longer represents the real world model and provides unhelpful values. As a comparison is being done between two models, many small factors can be ignored that are shared between the two systems, allowing for not all small deviations and factors to be considered.

The following assumptions and simplifications were made when converting real world systems into models to be simulated using computer software.

- **Time assumptions**
  - A night shift is assumed to begin at 22:00 and conclude at 06:00
    - Total simulation time of night operations is always 480 minutes
    - Varying T1 closures and late pickups of belly carts in the morning are excluded
  - Transfer time of belly carts when T1 reopens assumed to be immediate
- **Package assumptions**

  - Package routing percentages are assumed (XPS acceptable/SHC/rejected for current process, XPS direct accepted/additional human check/too early/sent to T3 for new process) (see Section 7.2.4 for assumed values)
  - Packages flagged as yellow by the system are assumed to be acceptable once the human checks them in the morning
  - No packages are assumed to be lost
- **Truck assumptions**
  - Drivers know how to use the interface and have no problems with its operation
  - All trucks take 20 minutes to unload all their cargo, regardless of whether they are unloading both general cargo and XPS at T3 or just XPS at T1
  - Any trucks still in the waiting queue when T1 reopens are removed from the system and not taken into account
  - The interarrival rate for trucks is assumed to be the same for every night shift and does not account for fluctuations in demand due holiday and seasonal influences
- **Other assumptions**
  - Two docks are assumed to be available at T3
  - Two T3 workers are assumed to be available to scan XPS packages into the belly carts
  - Two T1 workers are assumed to be available to scan XPS packages into the sorter
  - Scanning and performing acceptance checks of XPS packages when accepted at T1 is assumed to be 3 minutes a package
  - Scanning and performing acceptance checks of XPS packages when undergoing an additional human check at T1 after reopening is assumed to be 3 minutes a package
  - Scanning a package into the sorter is assumed to take 15 seconds (0.25 minutes)

7.2.3. Model limitations

As with any model replicating a real world process, it is not possible to capture all characteristics and aspects of a system. The simulation code simplifies several aspects of the process in order to create a manageable representation of the night-time delivery system of XPS packages. For example, the model assumes stable arrival patterns and average processing times, while in reality operational factors such as varying demand levels, staffing differences, seasonal fluctuations, and disruptions may impact system performance.

Despite these limitations, the model captures the primary operational attributes that determine the performance of the XPS package delivery process. Since the same assumptions and simplifications are applied consistently to both the current and proposed processes, the model remains suitable for evaluating the relative performance differences between the two systems and concrete and data-supported results can still be obtained.

7.2.4. XPS package classifications

Below are diagrams and explanations of the different package classifications which the models will use to distribute the XPS packages accordingly. As there is no official data on the true breakdown of these package types, estimations are assumed based on interviews with KLM Cargo staff and in-person observations.

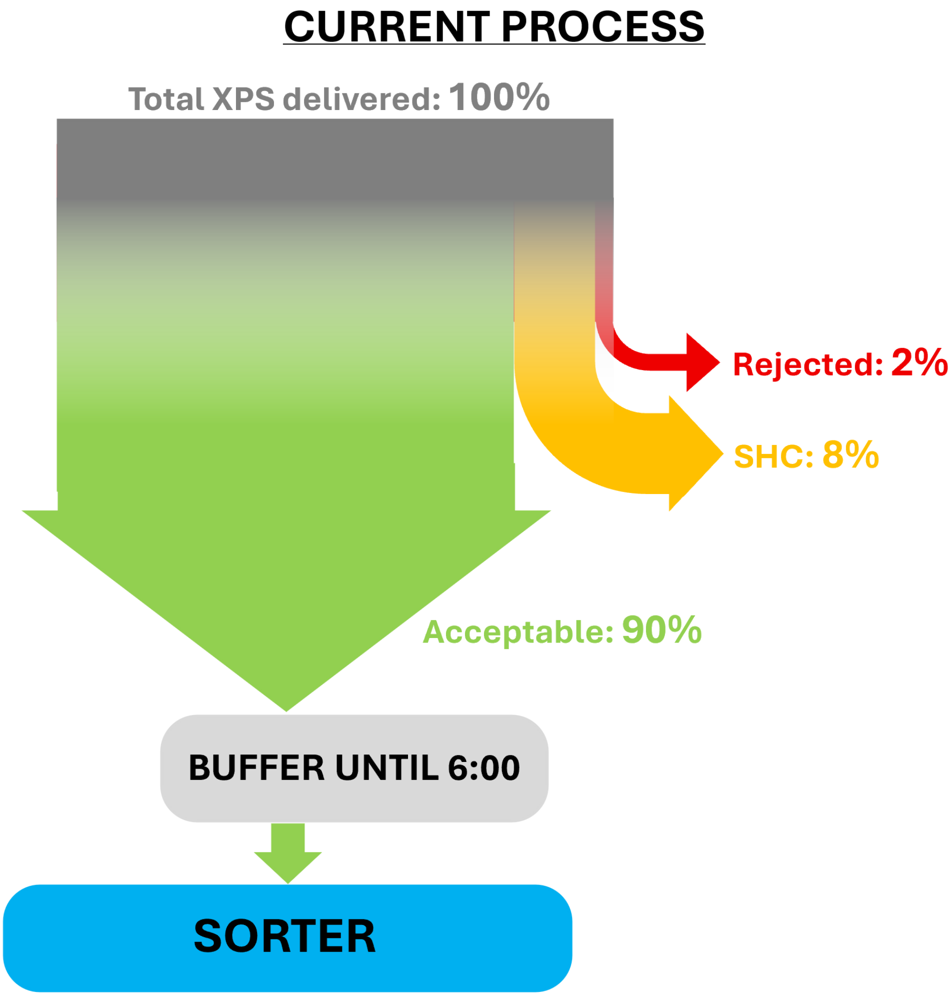


Figure 7.1: Distribution of XPS package types for the current process

The three types of package categories for the current process are as follows:

- Acceptable
  Assumed percentage: 90%
  These are XPS packages that have passed all acceptance checks performed by the T3 warehouse workers and have been marked "Ready for Carriage" (see 5.4.6). These packages do not require any special handling, have all paperwork and documentation correct, and are in good physical condition and may be inserted into the sorter. All acceptable packages delivered at night must wait in belly carts (buffer) until T1 reopens at 6:00 before being inserted into the sorter.
- SHC
  Assumed percentage: 8%
  These are XPS packages that have passed all acceptance checks but have a special handling code indicating they require unique handling depending on the item (e.g. dangerous goods, express heavy, cold storage, human remains, etc.). These may not enter the sorter and are handled via other means.
- Rejected
  Assumed percentage: 2%
  These are XPS packages that for some reason have been rejected by the warehouse personnel at T3. The package may be damaged, have a leak, showcase suspicious signs of potential tampering or sabotage, or have its physical characteristics (weight, volume, number of pieces) not match its AWB. These are not accepted by KLM Cargo and given back to the driver.

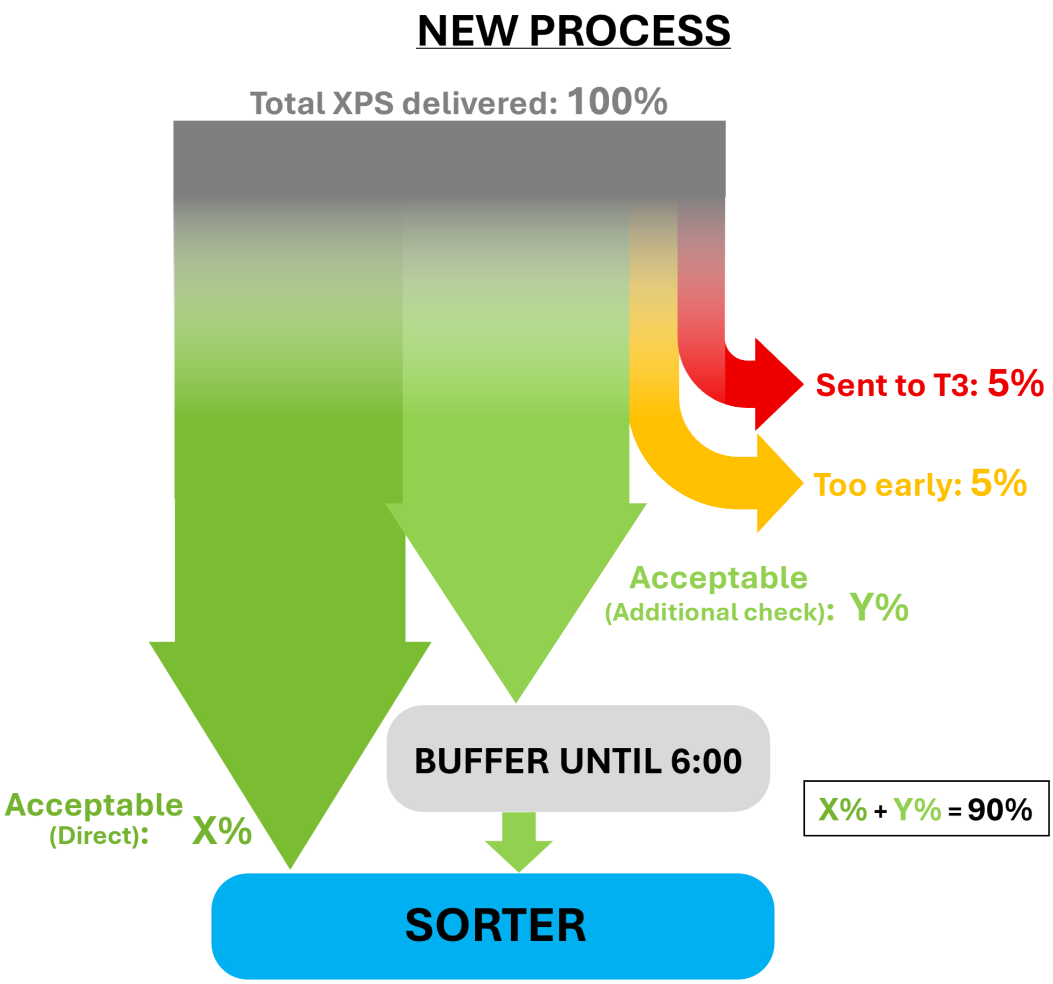


Figure 7.2: Distribution of XPS package types for the new process. What is important is the ratio between acceptable (direct) and acceptable (additional check) add to 90% to remain consistent with current process assumed percentage for optimal comparison

The four types of package categories for the current process are as follows:

- Acceptable (Direct)
  Assumed percentage: X% (where X% + Y% = 90%)
  These are XPS packages that have passed all acceptance checks performed by the automated systems of the new process and have been marked "Ready for Carraige" (see 5.4.6). These packages do not require any special handling, have all paperwork and documentation correct, and are in good physical condition and may be directly inserted into the sorter without needing to go into a buffer.
- Acceptable (Additional check)
  Assumed percentage: Y% (where X% + Y% = 90%)
  These are accepted XPS packages that the automated systems have flagged as yellow (see 6.4) and while do pass weight, volume, and number of pieces checks, have been marked as potentially being dangerous, suspicious, or damaged, and require additional verification from human warehouse workers at T1 in the morning. Therefore, all additional check packages delivered at night must wait in dedicated bins (buffer) until T1 reopens at 6:00 before being inserted into the sorter.
- Too early
  Assumed percentage: 5%
  These are XPS packages that have passed all acceptance checks performed by the automated systems of the new process and have been marked "Ready for Carraige" (see 5.4.6) but whose flight later that day does not have a chute ready yet. Thus, these packages are accepted but placed

in a holding bin for T1 workers to insert into the sorter later in the day at 12:00 when the chutes have opened up.

- **Sent to T3**
  Assumed percentage: 5%
  These are XPS packages that the automated system rejects and requires to be handled at T3 by human workers. These are packages without a proper IATA label, require special handling, whose credentials do not match whats on the AWB, that are with certainty determined to contain dangerous or illegal weapons, have leaks, or have clear indications of suspicious tampering. These packages are brought to T3 workers to be double checked by human workers to see if they are able to be accepted or not.

It is important to note that the exact proportion of XPS packages that can ultimately be inserted into the sorter is not known due to the lack of data. For the purposes of this study, it was therefore assumed that 90% of all arriving XPS packages are acceptable and able to enter the sorter for both the current and new processes. In the current process, all acceptable packages contribute directly to this 90%. In the new process, however, this proportion is divided between the direct and additional-check categories. By varying the distribution between these two categories while keeping their combined share equal to 90%, the performance of the new process can be evaluated under different operational scenarios and compared consistently with the current process.

Similarly, while the values for the other flows are also assumed, what is important is that they both add to 10% in both the current and new processes in order to have the percentage of XPS packages removed from the system be the same in both cases.

For the purposes of the model, the packages that have special handling codes or are rejected in the current process as well as the packages that are too early or need to be sent to T3 for the new process are considered to leave the system and are not modeled with the system focusing on the packages that are acceptable and insertable into the sorter.

7.2.5. Model overview

The fully detailed models for both the current and new processes are detailed below, with the former displayed in Figure 7.3 and the latter in Figure 7.4.

1. EVENT
Truck with **XPS packages** arrives at documentation and receives token

2. BUFFER
Truck with **XPS packages** drives to T3 and waits outside

3. EVENT
Truck with **XPS packages** called to dock and docks at the corresponding door

4. ACTIVITY
Truck fully unloaded (**XPS packages** + General Cargo)

5. ACTIVITY
XPS packages individually scanned, weighed, and checked by human

6. ROUTING
XPS packages sent to one of three locations
i. Accepted and sent to belly carts
ii. Requires special handling
iii. Rejected entirely

7.1a. BUFFER
XPS packages wait on belly carts

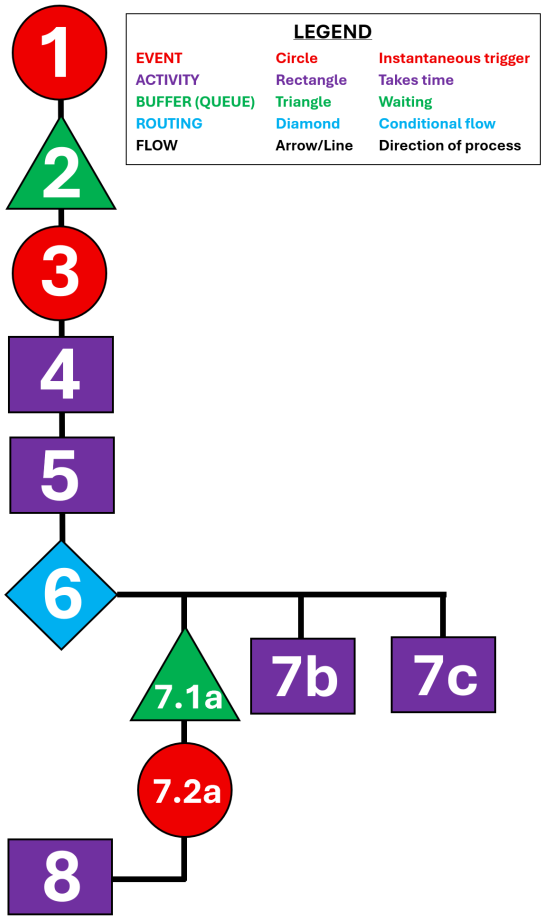
7.2a EVENT
XPS packages sent to T1 sometime between 6 and 7am

7b. ACTIVITY
XPS packages require special handling (XPH, DGs, pharma, cold storage, etc.)

7c. ACTIVITY
XPS packages rejected entirely and handed back to driver

8. ACTIVITY
XPS packages scanned one by one and inserted into the sorter

(a) DES components



(b) DES diagram

Figure 7.3: Current process representation

1. **EVENT**
   Truck with **XPS packages** arrives at documentation and receives token
2. **BUFFER**
   Truck with **XPS packages** drives to T1 and waits outside
3. **EVENT**
   Truck with **XPS packages** uses a drop off machine when its available
4. **ACTIVITY**
   Truck unloaded only of **XPS packages**
5. **ACTIVITY**
   XPS packages individually scanned, weighed, and checked by automated systems
6. **ROUTING**
   XPS packages sent to one of four locations
   i. Directly into sorter
   ii. Requires-additional-checks bins
   iii. Too-early bins
   iv. Not acceptable at T1, sent to T3

7.1a. BUFFER
XPS packages wait in require-additional-check bins until checked after 6:00

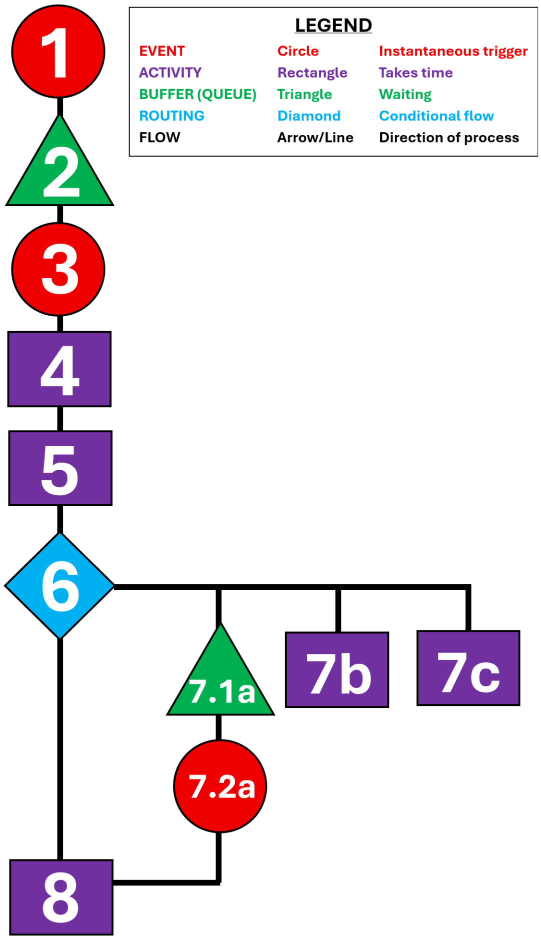
7.2a EVENT
T1 reopens allowing for **XPS packages** to be checked by humans

7b. ACTIVITY
XPS packages sent to too-early bins

7c. ACTIVITY
XPS packages rejected (SHC, damage, suspicious, etc.) and sent to T3

8. ACTIVITY
XPS packages sent into the sorter

(a) DES components



(b) DES diagram

Figure 7.4: New process representation

7.2.6. DES Structure

Building off the basic overview from the previous section, this section details the specific structure and order of the DES models for both the current and proposed systems. First, the system clock is explained, followed by a detailed event list with associated triggers, before ending with flow chart diagrams for the models.

System clock

The system clock begins the moment that the day operations cease and night operations begin. In the real world, this officially is to occur every night at 23:00; however, in practice, trucks bound for T1 are frequently sent to T3 earlier than this cutoff and thus the model follows this scenario and assumes the transfer occurs at 22:00. Instead of tracking the time literally, the simulation begins with t=0 representing 22:00 and increasing incrementally by whole minute until the last package is scanned into the sorter which represents the end of the simulation.

Event list

The following tables present the event lists for both the current and proposed scenarios, forming the foundation of the discrete-event simulation (DES) model. Each event represents a specific state change within the system and is defined by its corresponding function and triggering condition. Tables 7.1 and 7.2 below provide a structured overview of how system activities are initiated and how interactions between different processes are governed in both scenarios:

Table 7.1: Event list for current scenario

Event	Description	Trigger
Truck arrival	A truck is created and enters the system	Arrives based on a Poisson process
Truck parks at dock and begins unloading	A truck in the system will be called to a dock and then begin the unloading process of removing the XPS packages from within	When one of the two docks becomes available
XPS packages sent to T1	All the XPS packages temporarily stored in the buffer in T3 are sent to T1	After T1 reopens and all trucks have finished unloading and packages have been checked

Table 7.2: Event list for new scenario

Event	Description	Trigger
Truck arrival	A truck is created and enters the system	Arrives based on a Poisson process
Truck parks at drop-off machine and begins unloading	A truck in the system will park at a drop-off machine and then begin the unloading process of removing the XPS packages from within	When one of the two drop-off machines becomes available
XPS packages requiring additional check sorted	All XPS packages temporarily stored in the buffer in T1 are checked by T1 personnel and then sent into the sorter	After T1 reopens, this process begins

DES flowchart diagrams
For each process, a flowchart was developed to illustrate the movement of a truck and its contents through the system, including the routes, loops, and sequence of steps involved in the complete handling process. The flowcharts for both the current and proposed scenarios are presented below in Figure 7.5, highlighting the order of operations and the triggers that initiate events.

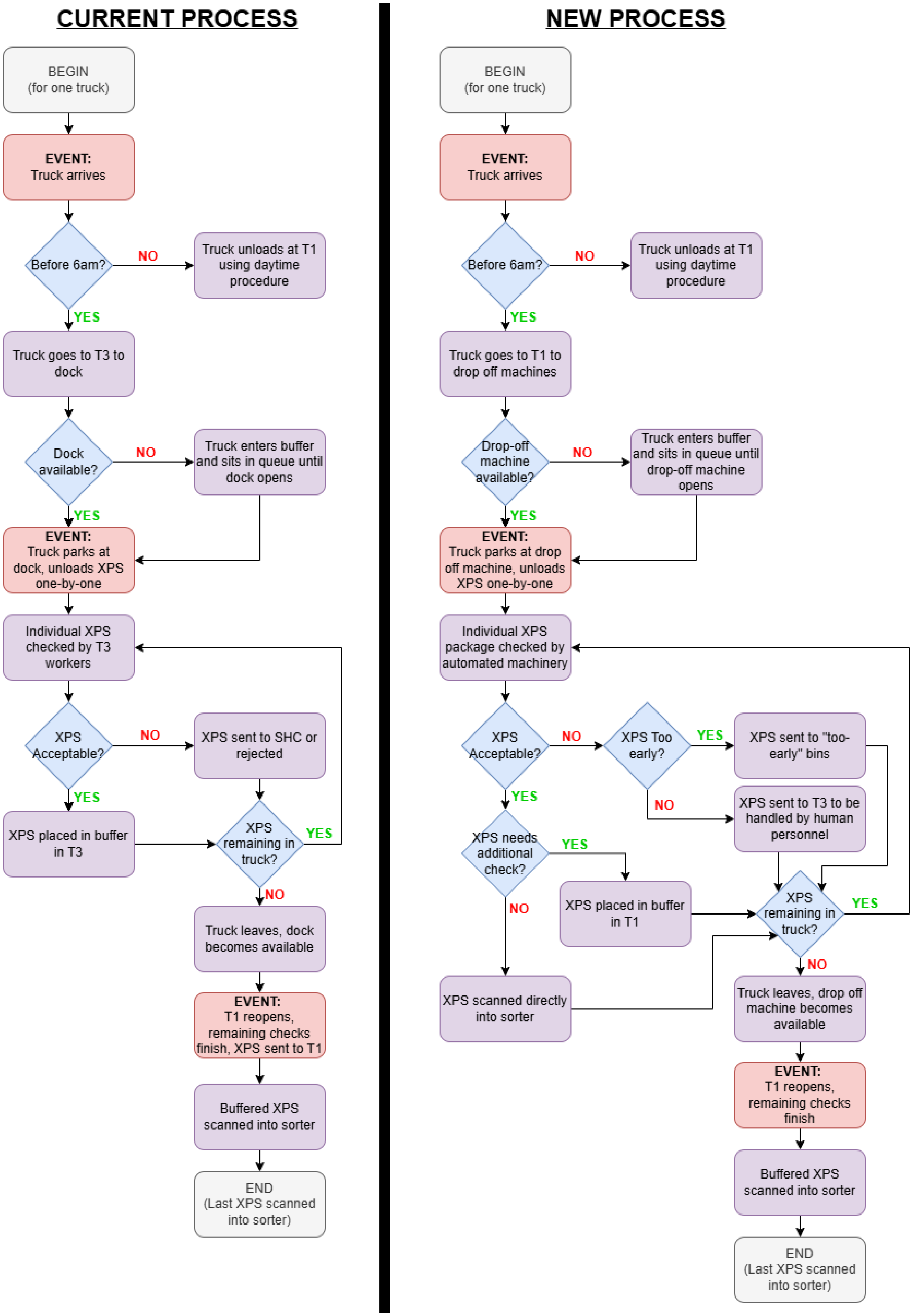


Figure 7.5: Flowchart diagrams of one truck and its respective contents for both the current and new processes

SUBQUESTION #5 - ANSWERED

In conclusion, the model for the project was chosen to be a DES due to the time-based nature of the package handling process and the ability to split the process in multiple steps with triggers, buffers, queues, and routing decisions. Two models were created to represent both the current process as well as the new process as closely as possible. Assumptions were made in order to help simplify the real world scenarios into a computer model.

7.3. Simulation

With both the current and the new process converted into a simplied DES models, they must then be converted into code and simulated in order to generate results that can then be analyzed. This section will cover which software and subsequent libraries were used when building and simulating the models in code as well as how the code and files were set up.

This section will answer the final subquestion written below:

SUBQUESTION #6 - TO BE ANALYZED

How can this theoretical model be simulated and used to evaluate and compare the performance of the proposed design against the current state?

7.3.1. Software details

The software chosen to convert the models into code and simulate them to obtain results was Python. The preexisting libraries that were chosen were numpy, simpy, pandas, and scipy.

7.3.2. Setup

The code is split into multiple Python files, some of which are shared by both the current and new process models with others being system specific. These will be described in detail below beginning with the files that are shared between the two before moving into the individualized files. The full code files can be found in Appendix C.

- **parameters.py**
  This file consolidates all the changeable variables of the two systems into one location enabling easy quick changes in parameters such as number of docks, unloading time, and package percentages
- **entities.py**
  This file defines the two objects that move through the system (Trucks and XPS packages) and stores their attributes
- **resources.py**
  This file creates the SimPy resources or areas where queues are able to form such as handlers, docks, or drop-off machines
- **kpi_tracking.py**
  This file comprises all the KPI functions and outputs the relevant KPIs for the
- **process_current.py**
  This file sets up the proper order of DES elements for the current system
- **process_new.py**
  This file sets up the proper order of DES elements for the new system
- **run_simulation_current.py**
  This file runs the current process environment by going through the current process steps, creating multiple runs, and determining the results

- **run_simulation_new.py**
  This file runs the new process environment by going through the new process steps, creating multiple runs, and determining the results
- **arrival_data.py**
  This file loads data from .csv files containing interarrival data and packages per truck data that will be used by the code when determining when trucks arrive and how many packages each will individually have

7.3.3. Simulation inputs and parameters

In order to run the code, various inputs derived from data and parameters chosen based on the real life process or on assumptions are needed. Below is a complete list of the inputs and parameters used in the model for both the current and new processes:

- **Inputs** - Stochastic values based on poisson distributions and historical data obtained from KLM Cargo datasets [16]
  - Truck interarrival times
  - XPS packages per truck
- **Parameters** - Fixed values
  - Simulation time: 480 min
  - Number of T3 docks: 2
  - Number of T1 drop-off machines: 2
  - Number of T3 workers: 2
  - Number of T1 workers: 2
  - Time T3 workers take to scan and check package: 3 min
  - Time T1 workers take to scan package into sorter: 0.25 min
  - Time T1 workers take to scan and check package: 3 min
  - Truck unloading time: 20 min
  - Current process package distribution percentages:
    * Acceptable: 90%
    * Special handling code: 8%
    * Rejected: 2%
  - New process package distribution percentages
    * Accepted direct: X% (where X + Y = 90%)
    * Requires additional-check: Y% (where X + Y = 90%)
    * Too early: 5%
    * Rejected and sent to T3: 5%

These parameters are easily modifiable and can be altered in order to test various other configurations and model scenarios.

SUBQUESTION #6 - ANSWERED

With nine python files coming together to form one comprehensive code that represents both the current and the new models, simulation was achieved in which KPIs could be obtained that allowed for a thorough comparison between both systems.

7.4. Validation

In order to ensure code properly matches up with the real world scenarios that they aim to replicate as well as confirm the results from Section 8.1 correctly reflect the processes that they are modeled after, a number of verification tests were set up and performed to validate the results.

7.4.1. Face validation

Face validation involves evaluating the model to assess how closely its logic and structure reflect the real-world process it represents. In both the current and new models, the key operational change points and triggers within the processes were identified and used as markers to define the primary steps of the system. The model was structured around the two main entities involved in the process which are the trucks and XPS packages. The first part of the model follows trucks as they arrive at the terminal, wait for dock availability, and complete the unloading process, while the second part follows the XPS packages as they are scanned, classified, buffered after unloaded, and ultimately inserted into the sorter.

Although some assumptions and simplifications were made to help reduce the complexity of the model so it could be more easily converted into code, these were primarily to eliminate irrelevant moments or steps or to remove factors and impacting sources that equally effected both processes and thus did not impact the results. What remained were two clear paths split into eight steps each for both the current and new process which matched the system order and path that trucks and packages take, emphasizing the primary differences between the two systems.

7.4.2. Input validation

The primary input for this model is the interarrival time of the trucks arriving at night containing XPS packages. Data on the arrival times of these trucks was collected by KLM for an eight month period which allowed for the determination of interarrival times between these trucks. These times were compiled into a list which clearly showed a poisson distribution (see E.1.6). The interarrival time data used for the input for the model is directly obtained from this data set and therefore the arrival behavior of the trucks in the simulation directly matches that experienced in the real world.

Additionally, the number of packages per airwaybill as well as the distribution of airwaybills per truck were also recorded over the same time period which together allowed for the determination of the distribution of the number of total XPS packages per trucks arriving at night. This data set was then used by the model where the model would randomly pick a value from this list when determining how many packages a truck would have, giving it a distribution that essentially matches what is observed in the real process.

7.4.3. Output validation

In the data produced by the models, select outputs can be used to validate the results by comparing them to real-world data. As shown below in Table 7.3, the outputs for the number of trucks that arrived per night as well as the number of XPS packages that arrived per night can be compared to the datasets that were provided by KLM Cargo (see E.5 and E.1 in Appendix E).

Table 7.3: Comparison of simulation-generated arrivals with observed KLM Cargo data

	Trucks per night		
	Simulation	**Observed data**	**Observed data (matching min)**
Minimum	6	0	6
Median	13.00	9	10
Mean	13.46	8.41	9.2
Maximum	23	18	18
Standard deviation	3.47	3.84	2.9
	Packages per night		
	Simulation	**Observed data**	**Observed data (matching min)**
Minimum	15	0	15
Median	57.50	16.5	29
Mean	64.88	23.16	33.4
Maximum	241	115	115
Standard deviation	38.02	21.42	19.6

The simulation results are compared both to the complete observed dataset (weekends excluded) and to a filtered version of the observed data in which only nights with a minimum number of arrivals are considered (matching the minimum obtained in the simulation). In both cases, the overall spread of the distributions is similar, but the simulation consistently produces higher values than the observed data. This indicates that the model captures the general behavior of the current, real-world process, while discrepancies arise due to simplifications and assumptions in the input modeling.

At first glance, the higher simulated values may appear problematic. However, this difference can be explained by several factors. Firstly, the historical dataset includes many nights during which no XPS trucks arrived, resulting in zero values for both trucks and packages. In practice, truck arrivals are influenced by external factors such as flight schedules, seasonal demand variations, and operational disruptions. These factors create a complex and time-dependent arrival pattern, including nights with no activity.

In contrast, the simulation does not explicitly model the occurrence of inactive nights with zero arrivals. Instead, arrivals are generated based on an interarrival-time distribution derived from observed data, which effectively represents nights with active operations. As a result, the simulated outputs reflect a higher average level of activity.

Secondly, truck arrivals in the simulation are generated using a Poisson process based on the observed mean interarrival time. This approach assumes a constant arrival intensity throughout the night and across all simulation runs. Consequently, it does not capture the variability in arrival intensity between different nights, such as peak periods, holidays, or low-demand periods observed in practice.

Finally, the simulation tends to overestimate the number of truck arrivals due to the method used to construct the interarrival-time distribution. The observed arrival times were converted into interarrival times by calculating the durations between consecutive truck arrivals within each night, excluding overnight gaps. However, this approach introduces a bias in which nights with many trucks contribute many interarrival observations, while nights with few or no trucks contribute few or none. As a result, the fitted interarrival-time distribution overrepresents busy nights and underrepresents quiet nights. When this distribution is used to generate arrivals throughout the simulated night, it leads to an overestimation of the number of trucks per night.

In addition, the number of packages per truck is sampled independently from a separate list obtained from a separate data sheet obtained by KLM Cargo. This removes any potential relationship between nightly demand levels and truck load sizes, which may further contribute to the overestimation of total package volumes.

Despite these differences, the variability of the simulated arrivals is comparable to that observed in the historical dataset. In particular, the standard deviation of the number of trucks per night in the simulation is similar to that of the observed data, indicating that the model captures the variability of the arrival process to a reasonable extent. Therefore, the simulation provides a suitable representation of truck arrivals for analyzing the performance of the night-time XPS handling process, while acknowledging the limitations introduced by the modeling assumptions.

8

Simulation results

In this chapter, the results obtained from which the overarching conclusions, discussions, and recommendations will be based are presented. These results obtained from these computer simulations are given. The chapter ends by discussing how the results were verified and validated, ensuring they match up properly with their real world counterparts.

8.1. Resulting outputs

In this section, the resulting data from running the simulation code is presented for both the current and new process scenarios. Additional charts and graphs that help display important comparisons are also provided.

8.1.1. Raw simulation outputs

When the simulation code was run for both the current and new processes, raw results were provided. This consisted of providing the raw data for each iteration run that the code ran (the number of iterations is selectable by the user and for this paper, was chosen to be 100 iterations) as well as a long list of averaged values across all iterations. Examples of the raw outputs by the simulation code for the current and new processes are shown in detail in section F.1.1 and F.1.2 respectively in Appendix F.

8.1.2. Refined simulation outputs for the current process

The current process was modeled to represent how the night-time delivery process of XPS packages is presently operated. However, the model reflects an optimized scenario in which only trucks containing XPS packages arrive at T3. Normally, T3 receives a large number of trucks carrying only general cargo during the night. Trucks carrying XPS packages compete for the same docks and personnel which therefore increases queue lengths and waiting times. Due to the lack of data on the arrivals and processing of general cargo trucks, they were not explicitly modeled. As a result, the current-process simulation represents an optimistic scenario in which XPS trucks are not hindered by any other truck types. Consequently, if the proposed new process demonstrates improved performance compared with this optimized version of the current system, it confirms that the new process offers a clear improvement over the existing approach. See below in Table 8.1 the results created by running the code file run_simulation_current.py with 100 iterations:

Table 8.1: Current process simulation results

System input	
Average trucks arrived	13,46
Average trucks processed	13,42
Average packages arrived	64,88
Average packages put in sorter	58,52
Overall system performance (min)	
Average mean XPS flow time	248,99
Average median XPS flow time	241,5
Average N95 XPS flow time	430,33
Average min XPS flow time	53,53
Average max XPS flow time	460,93
Average std. dev. XPS flow time	118,91
Process completion (min)	
Average first time XPS inserted in sorter	492,93
Average last time XPS inserted in sorter	500,06
Average total process time	500,06

8.1.3. Refined simulation outputs for the new process

The model for the new process allows the percentage of XPS packages routed **directly** to the sorter or sent to the **additional-check** process to be varied. As the exact operational distribution between these two routing outcomes is currently unknown, multiple routing scenarios were tested to provide a comprehensive overview of potential system behavior. The percentages are calculated out of 90% of the total packages, as approximately 10% of packages are assumed to be removed from the system due to factors such as being too early, requiring special handling (SHC), or being damaged. This 10% aligns with the assumed proportion of packages removed in the current process, allowing for direct comparison.

Table 8.2 presents the simulation results for ten routing scenarios, each evaluated over 100 iterations with all values averaged. Ten scenarios were selected to capture a representative range of possible routing distributions while avoiding unnecessary redundancy. The scenarios range from a best-case situation in which all packages are directly inserted into the sorter after processing through the drop-off machine (90%/0%), to the opposite extreme where all packages require an additional manual check the following morning (0%/90%). The intermediate scenarios represent varying distributions between these two routing outcomes.

Table 8.2: New process simulation results across varying direct/additional-check percentage combinations

	% Direct vs. % Additional Check									
	90/0	80/10	70/20	60/30	50/40	40/50	30/60	20/70	10/80	0/90
System input										
Average trucks arrived	13,26	12,19	12,67	12,8	12,36	12,2	12,39	13,39	13,65	12,98
Average trucks processed	13,06	12,07	12,57	12,65	12,2	12,06	12,15	13,15	13,46	12,89
Average packages arrived	63,43	60,46	56,49	60,66	55,68	53,92	59,04	63,57	64,65	61,09
Average packages put in sorter (total)	57,02	53,66	50,12	53,29	48,73	46,88	51,72	55,4	56,74	53,48
Average packages put in sorter (before T1 reopen)	52.08	42.93	34.46	31.75	25.16	18.99	16.70	11.80	5.53	0
Average packages (direct)	57,02	48,59	39,61	36,15	27,99	20,65	17,94	12,8	5,8	0
Average packages (additional check)	0	5,92	11,43	18,25	22	27,64	35,24	44,08	52,4	55,03
Direct XPS performance (direct) (min)										
Average mean XPS flow time	46,66	45,89	41,3	45,85	44,17	42,84	44,86	45,02	45,68	0
Average median XPS flow time	41,19	40,34	36,42	40,04	38,3	38,58	40,06	40,97	41,98	0
Average N95 XPS flow time	85,79	84,07	73,34	84,37	80,68	73,71	77,65	76,64	70,71	0
Average min XPS flow time	23	23	23	23	23	23,03	23,18	23,42	25,55	0
Average max XPS flow time	94,08	92,46	81,18	92,3	87,6	80,48	84,87	82,65	74,25	0
Average std. dev. XPS flow time	20,75	20,3	16,76	20,69	19,64	17,64	18,91	18,65	17,04	0
Additional check XPS performance (min)										
Average mean XPS flow time	0	240,82	246,78	263,05	271,45	283,44	264,46	290,96	291,65	283,61
Average median XPS flow time	0	242,55	246,25	253,28	269,89	278,31	258,69	296,69	292,38	284,06
Average N95 XPS flow time	0	324,32	367,84	394,15	401,93	413,87	407,3	427,33	434,25	420,74
Average min XPS flow time	0	143,26	122,53	123,03	127,19	131,7	114,97	122,26	125,7	127,29
Average max XPS flow time	0	334,42	391,09	415,41	427,5	438,12	432,46	446,94	458,7	447,58
Average std. dev. XPS flow time	0	73,81	93,22	94,01	94,02	93,64	98,4	98,73	98,13	95,09
Overall system performance (min)										
Average mean XPS flow time	46,66	66,55	83,19	115,35	139,87	176,69	187,12	232,57	267,57	283,61
Average median XPS flow time	41,19	43,59	42,99	58,59	79,23	151,18	179,82	247,96	275,77	284,06
Average N95 XPS flow time	85,79	214,89	300,1	350,07	367,39	402,05	393,87	420,14	431,92	420,74
Average min XPS flow time	23	23	23	23	23	23,03	23,18	23,42	30,27	127,29
Average max XPS flow time	94,08	343,23	392,29	415,41	427,5	438,12	432,46	446,94	458,7	447,58
Average std. dev. XPS flow time	20,75	71,31	96,49	118,45	130,88	141,76	134,39	137,82	120,19	95,09
Process completion (min)										
Average first time direct XPS in sorter	63,53	57,74	68,7	68,88	80,75	87,52	102,54	93,77	123,64	0
Average last time direct XPS in sorter	491,55	486,3	485,22	486,08	472,09	449,39	472,78	449	396,64	0
Average first time additional check XPS in sorter	0	405,72	468,51	483	483	483	483	483	483	483
Average last time additional check XPS in sorter	0	422,25	489,84	512,3	513,57	520,78	533,83	545,5	556,51	560,86
Average total process time	491,55	498,8	504,38	513,82	514,22	521,02	533,95	545,53	556,51	560,86

This table has been color coded to match up with Table 8.1 to allow for easier comparison. For each process, the system inputs are shown in **red**, the overall system performance is shown in **green**, and the process completion data is shown in **blue**. The performance results for the direct and additional-check XPS flows which are specific to the new process are shown in yellow and are not present in the current process results (due to all flow being sent into a buffer) but provide key insight into the difference between these flows and their potential. Additionally, they allow for comparing the

Although the exact operational distribution between the direct and additional-check flows cannot be determined due to limitations in available data on XPS package type distributions, the inclusion of multiple routing scenarios provides a wide representation of possible system configurations and performance metrics. Presenting the results across this range of scenarios establishes a foundation for evaluating how system performance varies under different routing conditions, enabling further analysis in subsequent sections.

8.1.4. Comparison table

Below is a table highlighting the changes in flow time between the current process and the variations of the new process. Each mean flow time is listed along with their respective improvement and percent increase/decrease.

Table 8.3: Comparison of average mean XPS flow time between the current process and the new process routing scenarios

Scenario	Mean flow time (min)	Δ Flow time (min)	Δ Flow time (%)
Current process	248.99	–	–
90/0	46.66	-202.33	-81.26%
80/10	66.55	-182.44	-73.27%
70/20	83.19	-165.80	-66.59%
60/30	115.35	-133.64	-53.67%
50/40	139.87	-109.12	-43.82%
40/50	176.69	-72.30	-29.04%
30/60	187.12	-61.87	-24.85%
20/70	232.57	-16.42	-6.59%
10/80	267.57	+18.58	+7.46%
0/90	283.61	+34.62	+13.90%

8.1.5. Comparison graphs and charts

The figures below compares the average XPS package flow time first between the current and the new process in Figure 8.5 and between the direct and additional-check flows of the new process in Figure 8.2. The flow time measures how long an individual XPS package remains in the system, defined as the time between the arrival of the truck carrying the package and the moment the package is inserted into the sorter.

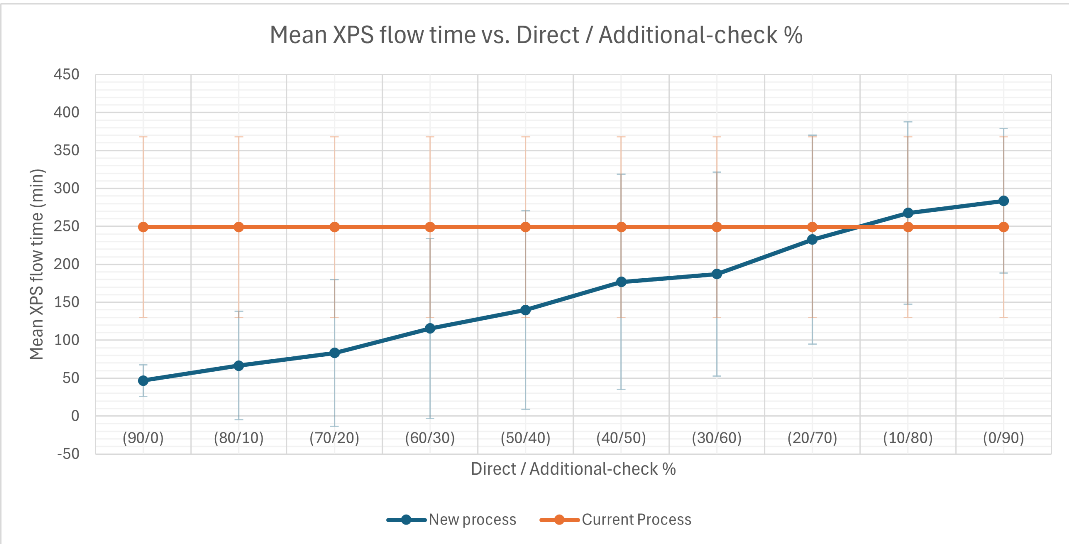


Figure 8.1: Average mean XPS flow time for both current process and various direct/additional-check percentages

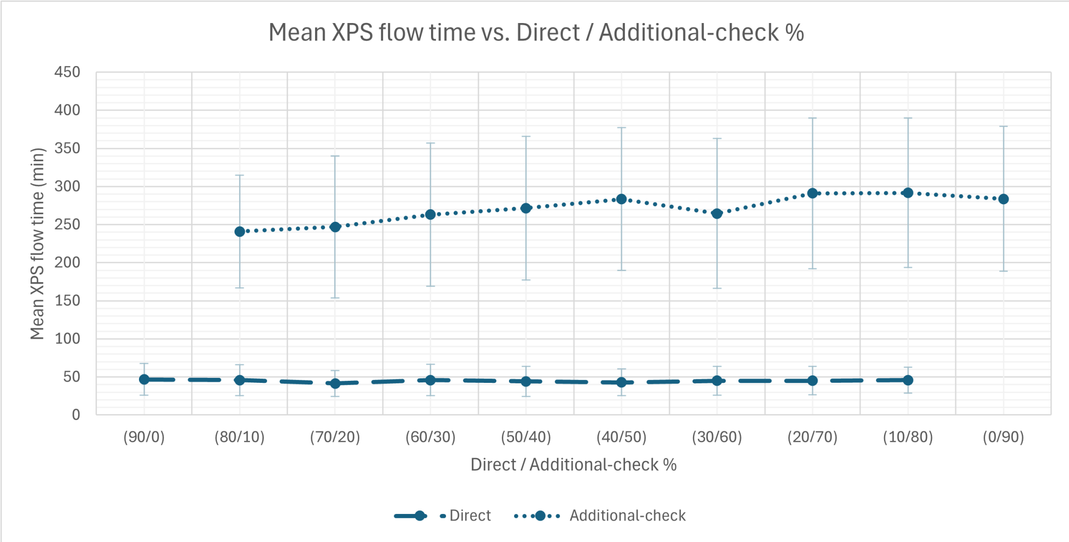


Figure 8.2: Average mean XPS flow time for direct vs. additional-check flows over various direct/additional-check percentages

The N95 flow time for both the current process and the ten different configurations of the new process are displayed below in Figure 8.3. The N95 flow time represents the 95th percentile of the package flow time distribution, meaning that 95% of all XPS packages spend less than this amount of time in the system.

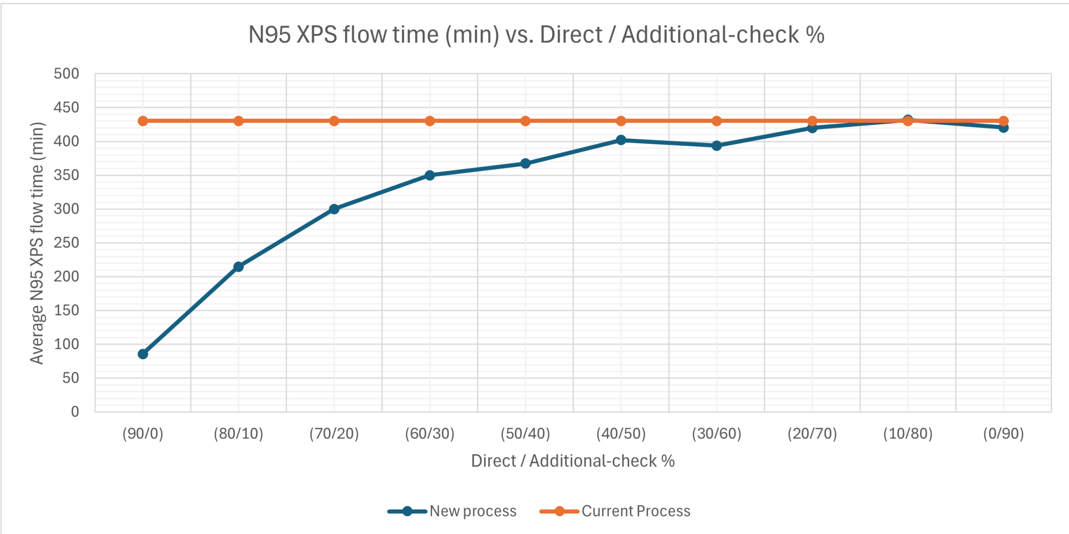


Figure 8.3: Average N95 XPS flow time for both current process and various direct/additional-check percentages

The figures below display the times when the very first and very last XPS package entered the sorter respectively. The new process is split between the direct and the additional-check flows to highlight the difference between the two types of flow.

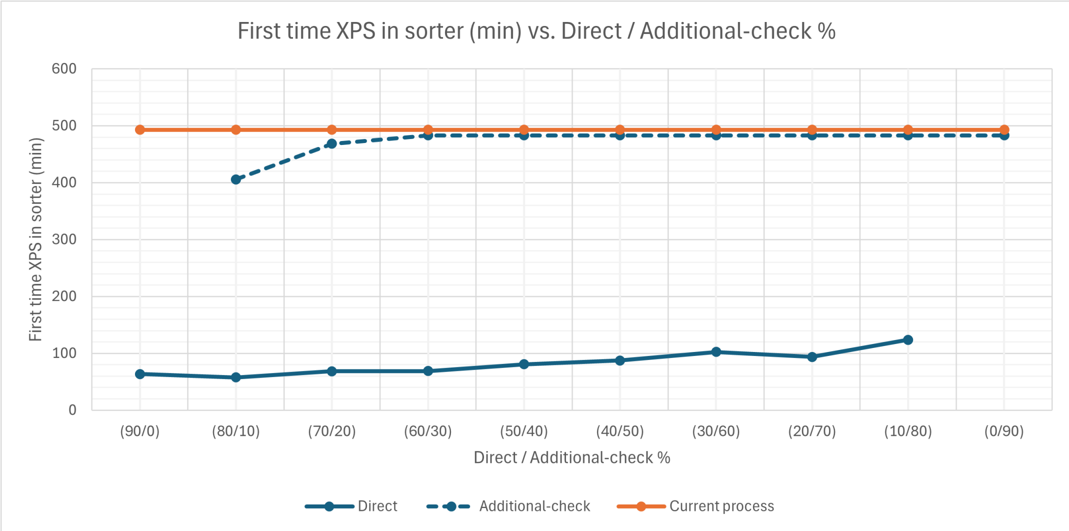


Figure 8.4: Average first time XPS package inserted into the sorter for both current process and various direct/additional-check percentages across the different flows and processes

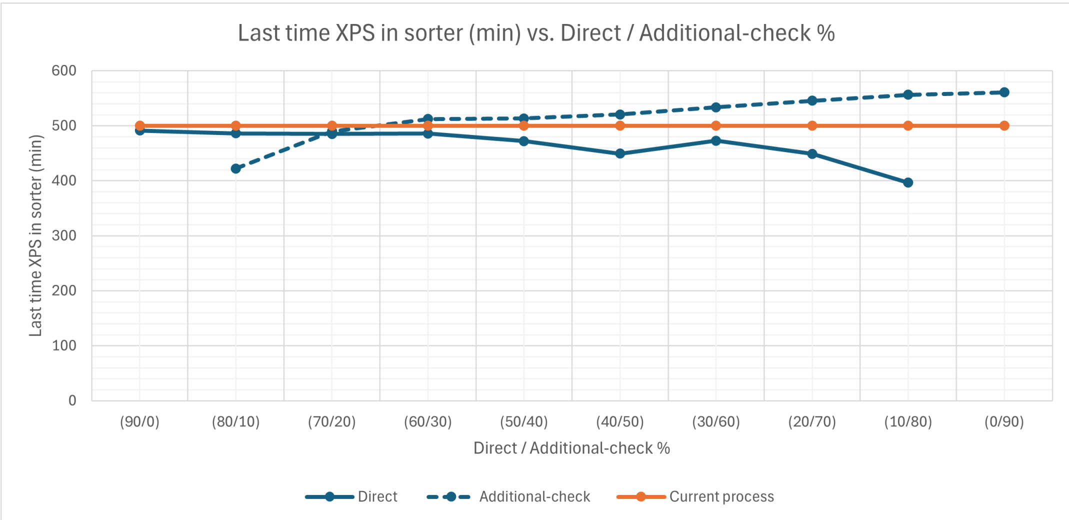


Figure 8.5: Average last time XPS package inserted into the sorter for both curremnt process and various direct/additional-check percentages across the different flows and processes

9 Discussion

In this chapter, the results and outputs from the simulations are discussed and compared between the two processes. We begin with the comparison of the static KPI metrics between the two before diving into the dynamic KPI results obtained from the simulations.

9.1. Static process metrics comparison

The static process metrics involve measurements of the process that do not vary or require models to compute. These include fixed distances and floor areas that can be calculated and determined using blueprints and can be corroborated with in-person observations. Section 4.4.1 discussed the metrics for the current process while Section 6.2.3 focused on the metrics for the new process.

In the current process, all XPS packages that are delivered at night arrive at T3, which as the crow flies is 440m from the Oogplein and the sorter where all accepted packages go. The path that accepted XPS packages take through T3, T2, and outside to get to T1 is 570m. In the new system, this distance is reduced to just 10 meters from the drop-off point to the end of the X-ray scanner when the packages then enter the sorter. This is a reduction of 98.25%. The paths for the Bulk flow and the Transito flow remaim unchanged and stay 121m and 187m respectively.

The floor area of T3 specifically used by belly carts on the far end of the voorloods was measured to be approximately 120 m^2 for the current process. This will no longer be needed due to the reduction in XPS packages that will be coming through the T3 voorloods, allowing this area to be free for other use. There may occasionally be a few XPS packages that the new system will flag as red and will be sent to T3 before being declared okay. These packages can then be sent to the belly carts at the DWARS location in T2.

The floor area of T1 will now be slightly reduced due to the implementation of the new system which is of a larger physical footprint than the current infrastructure that exists there which will be replaced. The required area will increase from the approximately $75m^2$ to $140m^2$.

In the current process, approximately one-third of night operations consist of XPS shipments, with the remaining two-thirds being general cargo (particularly during weekends). These relatively small XPS shipments are situated on large 1.2m x 0.8m europallets, typically with 1-3 packages per pallet. These take considerable amount of floor area for such a small product. With the new process aiming to remove all acceptable XPS shipments from T3 operations, this will drastically reduce the amount of floor area that is used for these large wooden pallets with only a few small XPS shipments on them.

9.2. Overview of simulation results

For all scenarios current and new, the average amounts of trucks seen per night as well as the number of packages per night was consistent with on average 12 trucks processed and 53 packages put in the sorter per night. Another metric that remained relatively constant was the final time that the last XPS was scanned, which was within the hour after T1 reopened at 6:00 (480 min).

The overall average time that an individual XPS package remained in the system decreased approximately linearly as the percentage of directly acceptable XPS packages increased. Additionally, the N95 flow time, representing the flow time below which 95% of all XPS packages fall, also decreased as the share of directly acceptable packages increased. However, this reduction occurred at an increasing rate as the system approached a near-perfect scenario of 90% direct acceptance and 0% additional checks, indicating that improvements in direct acceptance have an especially strong effect on reducing delays experienced by the slowest packages.

The times that the first XPS package entered the sorter remained constant for both the direct and additional check flows (the average for the additional check flows drop below 480 minutes in the (80/10) and (70/20) scenarios which appears to not make sense as these packages can only enter after T1 reopens so after 480 minutes, but some of the 100 run iterations happened to have zero additional check XPS which is why the average is brought down).

The times that the last XPS package entered the sorter also remained relatively constant for both the direct and additional check flows with the direct being slightly varying the lower its percentage went as less direct packages arriving meant more runs where there may be less direct packages delivered closer to the T1 reopening time. Additionally, the additional check times for last package inserted go up as their percentage increases due to the fact that the model has the morning checks for these packages take 3 minutes as they require extra time to ensure they are "Ready for Carriage" and to address the yellow flag where in the current process, once the packages are ready to be inserted into the sorter, they have already passed their inspection checks and so the model has them only take 0.25 minutes per package to be scanned into the sorter which is why the additional checked packages tend to finish later than those in the current process.

9.3. Performance comparison between current and new process

This section delves into the comparisons made between the current process and the many variations of the new process. First, the overall performance of each system is analyzed and compared before the impact of the varying percentage inputs for direct and additional-check XPS packages is examined. This is followed by an analysis of the process completion time as well as the number of XPS packages sorted before the night-shift ends. Lastly, the notion that the current process is actually the idealized version and normally is much less efficient and the implications behind that are also examined.

9.3.1. Overall system performance

The current process sees 90% of all delivered XPS packages be declared acceptable with the remaining 10% either being sent elsewhere to be specially handled or rejected and returned to the driver. For the purposes of modeling, they are removed from the system entirely. As they arrive throughout the night-shift, all packages within this 90% are stored in the buffer until the simulation time hits 480 minutes (when T1 reopens). If any trucks or XPS packages are still in the process of being unloaded or scanned, this finishes first. Once that is completed, they all then are scanned into the sorter, with the simulation time ending once the last package has been scanned into the sorter. This means that the time that the first XPS enters the sorter will always be after 480 min, regardless of when the package arrived to the facility. This keeps the average flow time for XPS packages in the system at a relatively high level of 248.99 min.

The new process consists of two flow types, direct and additional-check, in which the latter matches the way the current process is set up by sending packages into a buffer before having them be scanned in once T1 reopens at 480 min, while the former can enter the sorter directly and immediately after being successfully scanned and checked. In most cases, the average flow time of XPS packages in addition to the N95 XPS flow times were less than that of the current process. Additionally, while the first and last times that the additional-flow XPS packages were inserted into the sorter was similar to that of the current process due to both of them having to wait until 480 min had passed, the time that the first direct XPS was inserted into the sorter was significantly earlier than that of the current process.

As shown in Figure 8.2, the average time that a direct XPS package spends in the system is substantially less than the additional-check flows in all percentage combinations. As these packages no longer need to wait until the morning to be sorted and can be sorted right away after being checked by the automated systems, this significantly cuts down the time the package takes between arriving at documentation in the truck and being inserted into the sorter. The standard deviation for the direct flow is also much smaller, indicating a more precise and consistent process.

9.3.2. Impact of direct acceptance percentage

As it is not possible to determine the precise operational distribution between direct and additional-check flows due to the lack of available data on XPS package type distributions, multiple routing ratios were tested. This approach allows for an evaluation of how system performance changes as the proportion of directly accepted packages varies. By analyzing these scenarios, it becomes possible to identify the threshold at which the new process begins to outperform the current one, as well as the minimum proportion of directly accepted packages required to achieve improved performance. This provides a basis for assessing the feasibility of the proposed system and offers insight into the performance targets that would need to be met for successful implementation.

Ten different percentage combinations of these two flows were tested as explained in Section 8.1.3. The affect of the amount of directly accepted XPS packages vs. additional check XPS packages was very drastic and clearly showed a significant benefit to getting the former to be as high as possible. As the percentage of XPS packages directly accepted increased, the average time XPS packages spent in the system drastically declined in a linear fashion while the N95 XPS flow rates decreased in a seemingly exponential pattern.

The current system exhibited an average XPS flow time of 249.88 min which when compared to the linear results from the new process, matched up approximately with a direct/additional-check percentage split of (15/85). A direct percentage higher than 15% allowed for a lower average XPS flow time overall when compared to the current process, while a direct percentage lower than 15% caused the new process to perform less optimally and efficiently by having XPS packages remain in the system longer on average than the current process.

When comparing the performances of direct XPS packages with additional check XPS packages, the former has a clear reduction in time spent in the system in all regards. For all ten new process scenarios, not one XPS flow time result had a maximum time in the system higher than 95 min (which in of itself is quite high and usually is a result of when a truck arrives that is carrying a high amount of XPS packages which according to the data obtained by KLM Cargo and shown in the table in Section E.1.5 is unlikely but possible). The average flow time for direct packages is typically around 45 min with the shortest flow times being 23 minutes (20 minutes to unload the truck and 3 minutes to scan the package in). These values are incredibly low when compared to the additional flow statistics, in which the lowest minimum is just 122.53 min across all ten scenarios.

9.3.3. Process completion

The process completion time for all scenarios remained relatively constant due to the fact that in both processes, there exist packages that are stored in a buffer until morning, which causes the overall completion time to be sometime within the hour after T1 reopens. This number can be a bit deceiving, as even the most optimal setup of (90/0) has an average total process time of 491.55 min. This is to be expected though as XPS packages are delivered throughout the night so frequently trucks may still arrive between 5:30 and 6:00 in the morning whose XPS packages will need to be unloaded and handled which will last until after T1 reopens.

9.3.4. XPS sorting prior to T1 reopening

What the process completion time values do not show is how many packages have entered the sorter by the time T1 reopens. In the current process, no packages are sorted before T1 reopens as all packages must wait in the buffer before 6:00. For the new process, as the percentage of directly acceptable XPS packages goes up, the amount of packages that have been sorted prior to the reopening of T1 also increases. This is to be expected as since there are more direct packages in the system and these do not require waiting until the reopening of T1, more packages will have been sorted and taken care of by the time T1 reopens.

9.3.5. Suboptimal current process scenarios

This model simulates a idealized version of the current process in which the trucks carrying XPS packages experiences no hindrances, effects, or impacts from the many trucks that solely carry general cargo that are also being handled at T3. This correctly models what will happen at T1 in the new process as no other trucks arrive at this terminal during the night, but ignores the fact that many other trucks exist at T3 that the XPS carrying trucks need to wait for.

The table in Section E.1.7 shows the distribution of truck waiting times of trucks carrying XPS packages that arrive at T3 over an eight month period. This indicates that the waiting times for these trucks and in turn the XPS packages are far longer in real life than the model for the current process displays. The fact that the results for the new process are favorable when compared to the idealized version of the current process further strengthens the implications and conclusions that can be made when comparing the two as if the new process can be better than the perfect version of the current one, then it will be even more efficient and optimal than the scenarios in which the current process is impacted by these additional trucks.

9.4. Operational implications

In terms of real life operations of the terminals, the results indicate the following:

- **Truck waiting times and traffic congestion**
  The more packages that can be directly accepted into the sorter will allow for less trucks to need to go to T3, particularly the trucks that only contain XPS packages which are all either directly accepted or require additional checks. These trucks no longer need to travel to T3 and therefore reduce the overall congestion experienced there and the waiting times for the trucks that do need to go to T3.
- **T3 operations**
  With more packages being handled at T1 and less going to T3, the unloading of trucks and handling of XPS by T3 workers also is reduced. Less packages at T3 means less time and resources spent unloading, checking, and transporting them around. Additionally, floorspace will be saved as less XPS packages will occupy europallets in the T3 voorloods which frees up space and effort which can then be utilized for handling the general cargo that does need to use that space.
- **Packages missing flights**
  Having more packages be inserted into the sorter sooner will allow for less packages to miss their flights as they will already be waiting in their respective chutes ready to be loaded in the ULD or cart that will go to the plane at 6:00 as opposed to still sitting in a belly cart at the other end of T3 at the same time.
- **Lost packages**
  More packages already being sorted and not requiring a lengthy transfer process (where packages are also not properly tracked) will allow for less packages to be misplaced or lost as they will already be stored in their final destination directly after being handled by the new process.

9.5. Reflection on project objectives

The project objectives identified by the stakeholders are listed below. These objectives were previously defined in Section 5.2 of Chapter 5. For each objective, an explanation is provided to illustrate how the proposed process addresses these stakeholder requirements.

1. **Enable direct drop-off of XPS shipments at Terminal 1**
   The new process allows for all XPS shipments to attempt to be accepted at the new drop-off machines at T1. Whether they will be accepted depends on if the XPS package itself passes all criteria in order to be declared "Ready for Carriage" by the system. Packages that satisfy all prerequisites can be directly dropped off and inserted into the sorter.
2. **Improve operational efficiency and cargo flow**
   With all XPS packages declared "Ready for Carriage" and given a green flag by the new automated system able to be directly inserted into the sorter at T1, this will drastically reduce the number of XPS packages that need to be handled at T3, improving the operations of the export flow while

also expediting the XPS shipment flow as well.

3. **Improve prioritization and delivery reliability**
When an XPS package is accepted by the new system, it will automatically be sent to the chute for its intended flight which allows for high-priority packages to already be ready to go and be properly accounted for.
4. **Enable automated and retain 24/7 continuous operations**
The new system is designed to be entirely autonomous with human help only being required should a breakdown occur or a driver having difficulty. This system retains the ability to have XPS packages be deliverable at all hours while also now allowing for sorting to be 24/7 as well.
5. **Ensure regulatory compliance and operational safety**
The new X-ray machine, weight scale, dimension scanners, and AI computer software allow for checks of packages in the same way that human workers check packages at T3.
6. **Integrate seamlessly with existing systems and infrastructure**
Should the recommendation of utilizing Vanderlande for conveyor systems and screening technology be followed, this will make integrating the new system with the existing physical infrastructure much easier. Additionally, the new digital system will be directly linked with iCargo, allowing for much more real-time and accurate package tracking and data handling.
7. **Provide clear procedures and user-friendly system interaction**
The new process utilizes a clear and accessible interface for all users, with the ability to be able to use it in many languages. Additionally, help buttons and a call line to the documentation office is available should any questions or problems arise, allowing a connection to a human worker at all times.
8. **Ensure operational robustness and fallback procedures**
Should the entire drop-off system be down or undergoing maintenance, the current process will be the fall back temporarily until the new process is back up and running.
9. **Allow flexibility and scalability for future operations**
The system will relatively easily be able to be utilized during day operations as well, allowing for XPS drop-offs to be come entirely fully automated and allowing for T1 workers to focus on other duties and responsibilities during the day.

9.6. Limitations of the project

In any project that uses models to represent real world processes, limitations and hindrances to the data and results will exist. Several assumptions and simplifications were made when converting the real processes to computerized models. This was done when required data was not available to be used, when components of the system were shared between the current and new processes so could be ignored, or if it was too complex for the computer model.

For example, the unloading times of trucks were assumed to be constant when in reality they can vary for a variety of reasons, whether it be different speed capabilities of the drivers, the equipment available for unloading, the weights and dimensions of the packages, and how full the trucks were. Additionally, assumptions such as the drivers being able to perfectly understand and operate the new process interface as well as the machinery not experiencing any breakdowns or downtime were also made. Incorporating all these factors would have increased the complexity of the models significantly and made it more difficult to make comparisons between the two systems.

Although these simplifications can influence the results produced by the simulation and in turn potentially make them less accurate, they allow the model to focus on the relative performance differences specifically between the current and proposed processes, which is the primary objective of this project.

Additionally, a key limitation of this study was the lack of available data in two important areas. The first concerns the lack of information on the underlying reasons for shipments being classified as “departure not okay” (DEPNOK) versus “departure okay” (DEPOK). The former indicates that an express shipment did not make its intended flight, while the latter indicates successful departure as scheduled. As KLM Cargo aims to maximize DEPOK rates to improve reliability and customer satisfaction, understanding the causes of DEPNOK outcomes is vital. Although data was available on which shipments were classified as DEPNOK or DEPOK, as well as their intended and actual flights, no information was provided on the reasons behind delays or missed departures. Consequently, it was not possible to directly link these outcomes to inefficiencies in the current night-time delivery process or evaluate how the proposed system would impact them.

The second limitation relates to the lack of data on the distribution of XPS package types and associated special handling codes. Apart from XPH shipments, no detailed information was available on the type of XPS packages or the distribution of special handling codes, such as dangerous goods. As a result, it was not possible to determine the actual percentages of packages that would be accepted, rejected, or sent to be handled elsewhere. This required the distributions to be assumed (as described in Section 7.2.4), which still allowed for comparisons between the performance of the current process with that of the new, but limited the ability to accurately quantify the true distribution of packages across these flows.

10 Conclusion and recommendations

In this chapter, the final conclusions based on the simulation result and discussions are presented. Additionally, scientific and practical recommendations for both KLM Cargo and external parties are provided along with suggestions and opportunities for future research and development.

10.1. Conclusion

The goal of this project was to develop a new night-time cargo delivery process for XPS packages at KLM Cargo's Terminal 1 that would improve upon the efficiency, reliability, and operations of the current existing process while satisfying safety and stakeholder requirements. A proposed design was developed based on investigative research involving XPS packages being accepted at T1 in an automated drop-off system. DES models were created to represent both the current and newly proposed processes in order to compare the outputs and efficiencies between the two processes. The results obtained from the simulations allow for conclusions to be made as to whether the new proposal would be beneficial and how to best improve upon the current situation.

10.1.1. Summary of key findings

The results from the DES simulations clearly showed that the new process consistently performs better than the current process. Only when the percent of directly acceptable packages fell below 15% did the average XPS package flow time rise above that of the modeled current process which is very low. If over 15% of packages can be accepted directly into the sorter, then the new process already will exceed the performance of the existing process, according to the data from this report.

Additionally, the model representing the current process is basing it off of its most optimized version where it is not impacted by trucks carrying general cargo at T3. Should this be taken into account, the average flow times for XPS packages would be significantly higher in the current process due to increase truck waiting times for when a dock becomes available. Therefore, the results showing the improvements that the new process makes over the current process are actually less than in reality, as the latter is being compared to the fully optimized version of the former, as opposed to a more average night-shift, indicating the operational benefits fo the new process are actually even more substantial than they are being presented to be.

10.1.2. Main insights and implications

The largest improvement that the proposed solution design makes is allocating a percentage of the flow of XPS packages destined to the sorter to be directly inserted right after being deemed "Ready for Carriage" as opposed to having all of the packages need to wait in a buffer which is arguably the largest bottleneck in the XPS acceptance process and contributes the most to the high process flow times. The higher this percentage of directly accepted XPS packages, the fewer the number of packages that need to wait in the belly carts of T3, and the higher the overall efficiency of the entire process.

In addition to an overall lower average flow time for XPS packages in the system, this process will allow for more packages to be removed from needing to be handled at T3 during the night, which boasts several subsequent improvements in of itself. A lower amount of XPS packages that go through T3 means that less trucks need to go to T3, lowering traffic congestion and reducing truck waiting times for the trucks that do need to go there. Additionally, less floor space and resources will be needed when trucks are unloaded. The belly carts that stored temporary XPS packages in T3 can now also be removed, further opening up the voorloods floor space. Lastly, more XPS packages will be immediately ready for their flights in the morning as they will already have been sorted by the time T1 reopens, allowing for fewer missed flights and delays overall.

10.1.3. Research question answer

The main question that this thesis aimed to answer is as follows:

> **PRIMARY RESEARCH QUESTION**
>
> How can the delivery process of express packages during nightly closures of T1 be redesigned to optimize flow, improve operational reliability, and minimize transport time and distance, while complying with relevant safety and stakeholder requirements?

This primary question was successfully answered by creating and simulating a new solution design for the night-time delivery process of XPS packages which involves an automated drop-off system built into the side of and within the voorloods of T1. This new process will be able to autonomously accept a portion of the XPS packages delivered at night, check that they follow all compliance requirements, and directly sort them into the proper chutes of the sorter machine in T1. Additionally, it will temporarily store other XPS packages that require an additional verification check by warehouse workers to ensure that safety and security checks are still fully performed.

This split in flows allows for a reduction in the average flow time for XPS packages in the system, even if just 15% of accepted packages are able to be directly inserted into the sorter at night. Additionally, the total distance that XPS packages must travel also decreases on average as the shipments are delivered to T1 directly as opposed to T3. This shift of delivering them to T1 also successfully satisfies many stakeholder requirements such as reducing the effects of accepting XPS packages on T3 operations, ensuring less packages get lost, increasing tracking and priority sorting, and having more packages make their flights in the morning.

10.1.4. Research gap filled

The official gap definition, as previously determined in Section 2.4.1, is as follows:

> **GAP DEFINITION**
>
> A notable absence exists of focused analysis on cargo facilities where some terminals close temporarily while others remain continuously operational.

While this report focuses on increasing the overall efficiency of cargo terminal facilities, what is unique is it focuses on a terminal layout in which one terminal closes temporarily while the others remain open. This operational schedule is shared by other airports and cargo terminals around the world but little research is performed on these setups, causing these existing locations to use localized and improvised processes.

This project allows for the gap to be partially filled as it presents a proposed scenario that is shown to benefit this kind of terminal situation which can then be utilized as a basis for other airports and terminal situations around the world, as well as be the basis for future terminal facilities that do not exist as of yet but may potentially want to use this setup.

10.2. Recommendations

Based on the results obtained from the DES models and the analysis of both the current and proposed processes for night-time delivery of XPS packages, several recommendations can be made for both the scientific community and the stakeholders involved, should the proposed solution design be implemented. Additionally, remarks on the opportunities for future research and potential next steps are also provided.

Below are the recommendations suggested to the parties most affected based on the data and results from this project.

10.2.1. Scientific recommendations

As mentioned in Section 10.1.4, this project helped address the gap in the scientific field which focuses on analyzing and improving terminal facilities with partial scheduled closures. It is recommended that introducing standardization and guidelines for implementing automated systems such as those for package drop-off at terminal complexes such as that found at KLM Cargo is vital due to the efficiency gains that are projected to be obtained from introducing this new process.

Additionally, this project showcases the benefit of using DES modeling in a multi-cargo terminal layout due to the reduction in average flow times and increase in packages being automatically sorted. It is therefore recommended that DES be used in similar situations due to its ability to replicate processes well and obtain verified and crucial results.

Outside of the research gap, the new process also delves into processes that are partially fully autonomous while also still retaining a meaningful level of human interaction. This is useful to present as there exists significant hesitancy in implementing AI and automation in processes and what the implications are of that.

10.2.2. Practical recommendations for KLM Cargo

Based on the results of the project, it is recommended to KLM Cargo to invest in an automated drop-off solution for accepting XPS packages during the night when T1 is closed. The newly proposed design is projected to provide significant time saving gains for XPS packages being sorted after delivery while also improving the efficiency of cargo flow, reducing congestion, and mitigating delayed or missed flights.

It is recommended to have a minimum of two drop-off machines incorporated into the design as that way if one is out of order for any reason, the other will still be able to be used and the system can continue without needing to fully shut down.

Additionally, it is highly recommended to utilize Vanderlande in the physical construction of any new system as they already are the sole company responsible for the existing conveyor systems across all of Schiphol and retaining them will allow for a more streamlined and consistent integration process and eliminate having to consult and manage two separate systems together.

10.2.3. Practical recommendations for external parties

While this project was presented and overseen by KLM Cargo, its applications and conclusions can be valuable to external parties beyond the company as well. Any other logistical setup that takes in deliveries of small packages during off hours can benefit from this new design. Situations like cargo terminals at other airports, package distribution centers such as those used by Amazon or DHL, post offices, or even retail stores that receive returns or deliveries will be able to benefit from such systems. More research will have to be conducted in order to ensure the scale and output of the process matches the specific requirements of the party in question, but this report recommends that using automated drop-off technology can save time and increase organization and efficiency in the package delivery process.

10.3. Future research and next steps

While this report provides substantial evidence of the efficiency gains and operational benefits that the newly proposed night-time delivery process will provide for KLM Cargo, there are many areas that would benefit from additional investigative research and analysis. This is particularly relevant given the assumptions made and the elements that were beyond the scope of this project, which could be further investigated through additional research and testing. These are explained in the following list:

- **Automated compliance checks**
  One of the most important aspects of cargo transport is ensuring cargo is safe, secure, and compliant Specific guidelines are in place (see 5.4.6) to ensure that all packages that KLM Cargo meets satisfy these requirements. Currently, these checks are performed by humans but the new process proposes these checks be done by autonomous technology which has been increasingly become common place in similar industries and warehouses.

  The interview with the KLM Cargo staff member working in the company's Compliance and Safety (C&S) department uncovered a relatively high level of hesitancy and doubt that AI technology will be able to fully complete these checks as well as they currently are being performed by humans. Should KLM Cargo want to invest in this project, it is recommended that additional research be performed to ensure that the safety and security of cargo acceptance can be as or even more reliable than manual checks.
- **Classification of XPS packages**
  Currently, there was no data available from KLM Cargo on the distribution of the classifications of XPS packages that arrived at the premises, regardless of the time. It would be beneficial to the company to invest in more research and data collection to be able to discern which packages are of which types and special handling codes as this will allow for more accuracy in determining package flows and XPS distributions.
- **Importance of X-ray scanning**
  There are presently no X-ray machines on the premises of KLM Cargo as all cargo that arrives is delivered by "secure" clients and thus this is not needed. The suggestion of incorporating an X-ray machine in the new process was made due to the process now being fully automated and therefore would allow for increased security in the absence of human checkers. However, additional studying and research can be performed in order to see if this step is deemed necessary for KLM Cargo's operations or if it can be excluded as this would save in time and floor space for the new design.
- **Impact of system downtime**
  The models in the report assume no breakdown time for any of the process's machines or equipment. Further research can be performed now to analyze the potential impacts that system downtime, failures, or maintenance intervals will have on future operations.
- **Modeling the current system with T3 trucks**
  The project also assumes the current process at T3 is in its most optimal form in which it experiences no hindrances from trucks carrying only general cargo which frequently arrive during the night and cause high truck waiting times. While this does not affect the results of this project, more research can be performed to develop an even more accurate representation of the current process which can be used in other studies.
- **Automated drop-off for day operations**
  The scope of project only focused on the night-time operations at KLM Cargo when T1 was closed. In the interviews, it was determined that there was significant interest in also applying such a system for day-time operations as well if the process was shown to improve night-time operations. As this was the case, it is recommended that more research now be undertaken to see the positive and negative impacts that such a system would have on day-time operations as well.
- **XPS with SHC**
  Both the current system and new process reject XPS packages with special handling codes. Due to the ever increasing amount of automated systems being incorporated in logistic environments, further research can be performed in order to see how automated systems could best handle packages that presently require human intervention such as dangerous goods, human remains, and cold storage.
- **New cargo terminal**
  With KLM Cargo designing and developing a single unified new cargo terminal for its operations

destined to open in the 2030s which would render this setup of partially closing terminals obsolete, more research and studies can be done to see how automated drop-off systems for XPS packages would be beneficial for this new terminal setup as well.

References

[1] Air France KLM Martinair Cargo. *Product Portfolio 2025*. Company product overview document describing cargo services and requirements, including express and express heavy shipments. Air France KLM Martinair Cargo, 2025. URL: https://www.afklcargo.com/MY/en/products_and_solutions/products_and_solutions (visited on 03/04/2026).

[2] Athanasios Ballis. "Overview of air cargo terminal design aspects". In: *Transportation research record* 2007.1 (2007), pp. 117–125.

[3] Hilde Blakenburg et al. "AI-supported customs controls for the flow of defense and dual-use goods". In: (2026).

[4] Elvira Borregana. *Express zending tijdens de nachtdienst aannemen in de voorloods van VG3*. Werkinstructie. Version 1. Owner: Rick van der Waard, internal operational document. KLM Cargo. Aug. 5, 2024.

[5] Marcel Bos. *Dwarstransport VG1 en VG2/VG3*. Werkinstructie. Version 1. Owner: Tim van Straten, internal operational document. KLM Cargo. May 27, 2024.

[6] Ching-Cheng Chao and Po-Cheng Yu. "Quantitative evaluation model of air cargo competitiveness and comparative analysis of major Asia-Pacific airports". In: *Transport Policy* 30 (2013), pp. 318–326. ISSN: 0967-070X. DOI: https://doi.org/10.1016/j.tranpol.2013.10.001. URL: https://www.sciencedirect.com/science/article/pii/S0967070X1300156X.

[7] H. Deben. *Noodscenario palletizer op oogplein*. Operationele Instructies. Version 1. Validated by Shiftleader EPS, internal operational document. KLM Cargo. Aug. 18, 2020.

[8] *Frankfurt Airport Nightly Closure: Economic Impact*. 2020. URL: https://ivypanda.com/essays/frankfurt-airport-nightly-closure-economic-impact/?utm.

[9] Mehmet Gurturk and Erkan Celik. "A Bibliometric Analysis of Air Cargo: Milestones, Trends, and Future Directions". In: *Transportation Journal* 64.2 (2025), e70001.

[10] Seock-Jin Hong, Woongyi Kim, and Brian Hiatt. "Examining airport agility at air cargo hub airports". In: *Journal of Air Transport Management* 122 (2025), p. 102710.

[11] International Air Transport Association. *Abbreviations and Special Handling Codes*. Section 1.3. International Air Transport Association (IATA). Montreal, Canada, 2010.

[12] II Kleshko et al. "Automated baggage screening and logistics system for enhanced airport efficiency". In: *E3S Web of Conferences*. Vol. 592. EDP Sciences. 2024, p. 07003.

[13] KLM Cargo. "Accept Shipments as Ready-for-Carriage". In: *Cargo Handling Manual (CHM)*. Issued by SPL/FQ, internal operational manual. Dec. 18, 2025. Chap. 8.

[14] KLM Cargo. "Content, Terminology Training". In: *Cargo Handling Manual (CHM)*. Issued by SPL/FQ, internal operational manual. Dec. 18, 2025. Chap. 1.

[15] KLM Cargo. "Express Shipments Including DIP & LHO". In: *Cargo Handling Manual (CHM)*. Issued by SPL/FQ, internal operational manual. Aug. 1, 2025. Chap. 16.04.

[16] KLM Cargo. *Operational datasets for cargo handling and truck arrivals*. Internal operational datasets provided by KLM Cargo for thesis research. Includes truck arrival times, docking times, waiting times, and express shipment characteristics. 2025.

[17] KLM Real Estate & Facility Contracting. *Plattegrond Begane Grond – Gebouw 545 Vrachtgebouw 1*. Project: Vastgoed, Veiligheid en Milieu; Object: Gebouw 545 Vrachtgebouw 1; Drawing number: 11031; Internal engineering drawing. Sept. 10, 1997.

[18] KLM Real Estate & Facility Contracting. *Plattegrond Begane Grond Laag-A – Vrachtgebouw 3 Schiphol Centrum*. Project: Bordessen en roostervloer; Object: Vrachtgebouw 3 Schiphol Centrum; Drawing number: B26920P0; Internal engineering drawing. Sept. 16, 2014.

[19] Dethpon Lapphiphitmongkhon. "Improving air cargo handling operation: the case of an international airlines in Taiwan". In: (2008).

[20] Praveen Nath S and Ram Krishna Upadhyay. "Reformation and optimization of cargo handling operation at Indian air cargo terminals". In: *Journal of the Air Transport Research Society* 2 (2024), p. 100022. ISSN: 2941-198X. DOI: https://doi.org/10.1016/j.jatrs.2024.100022. URL: https://www.sciencedirect.com/science/article/pii/S2941198X24000332.

[21] Francess Chinyere Okolo et al. "Strategic framework for enhancing cargo screening and intelligent border security through automated detection technologies". In: *Journal Of Frontiers In Multidisciplinary Research* 3.1 (2022), pp. 150–159.

[22] Nisara Paethrangsi, Chadaporn Jaktong, and Parida Brahmaputra. "Determinants Affecting Air Cargo Operational Performance in Thailand: A Strategic and Mixed-Methods Approach". In: *Interdisciplinary Academic and Research Journal* 5.1 (2025), pp. 1023–1042.

[23] Michail Papoutsidakis, Aikaterini Sfyroera, and Abhishek Srivastava. "Case study of automated baggage handling in modern transportations". In: *Int J Eng Appl Sci Technol* 4 (2019).

[24] Deni Priansyah et al. "Planning of Cargo Terminal Area Requirement of Komodo Labuan Bajo International Airport". In: *Advances in Transportation and Logistics Research* 7 (2024), pp. 839–853.

[25] Gregor Selinka, Axel Franz, and Raik Stolletz. "Time-dependent performance approximation of truck handling operations at an air cargo terminal". In: *Computers & Operations Research* 65 (2016), pp. 164–173.

[26] Abdul Salam Shah et al. ""Enhanced airport operations: Automated baggage drop-off and boarding pass generation for travelers,"" in: *Journal of Advances in Technology and Engineering Research* 10.2 (2024), pp. 1–6.

[27] Ravi Singh. "Innovative baggage handling solutions to enhance passenger experience". In: *Journal of Airport Management* 18.1 (2023), pp. 18–31.

[28] Richard Smith. *EXPRESS-aanlevering VG03*. Lokale procedures Schiphol. Version 2. Validated by Shiftleader Depts, internal operational document. KLM Cargo. Mar. 30, 2020.

[29] Bram Talahatu. *Oogplein afhandeling*. Operationele Instructies. Version 3. Validated by Shiftleader EPS, internal operational document. KLM Cargo. Feb. 12, 2021.

[30] Yashraj Thakur et al. "Automated Baggage Storage System". In: (2025). Ed. by Vinit Kumar Gunjan and Jacek M. Zurada, pp. 479–487.

[31] Han Win Thein. "A Study on Cargo Operation Management in Sky Cargo Terminal (Han Win Thein, 2025)". PhD thesis. MERAL Portal, 2025.

[32] Agnieszka A Tubis, Anna Jodejko-Pietruczuk, and Tomasz Nowakowski. "Use of augmented reality as a tool to support cargo handling operations at the CARGO air terminal". In: *Sensors* 24.4 (2024), p. 1099.

[33] Jaroslav TUREČEK, Tomáš HOIKA, and Zbyšek KORECKI. "INCREASING THE EFFICIENCY AND CAPACITY OF AIRPORT SECURITY CHECKING BY ARTIFICIAL INTELLIGENCE." In: *Transport Problems: an International Scientific Journal* 20.2 (2025).

[34] Theo Viejou. *Pakketten met gemiste vluchtverbinding*. Operationele Instructies. Version 1. Validated by Shiftleader EPS, internal operational document. KLM Cargo. Aug. 31, 2020.

[35] Danijela Vukadinovic, David Anderson, et al. "X-ray baggage screening and artificial intelligence (AI)". In: *JRC science for policy report, EUR* 31123 (2022).

[36] IMT Wei. "Improving the air cargo terminal operations through optimization". PhD thesis. Master's thesis. National University of Singapore, 2005.

A

Appendix A

A.1. Scientific paper

Night-time Cargo Delivery Process at KLM Cargo

D.N. Bravo[1,*] Dr. J.M. Vleugel[1] Dr. P.S.A. Stokkink[2]

19 April 2026

[1] TU Delft – Faculty of Civil Engineering and Geosciences
[2] TU Delft – Faculty of Technology, Policy, and Management
[*] Corresponding author

Abstract

Abstract

Cargo handling is a continuous 24/7 operation. However, at KLM Cargo's facilities at Amsterdam Airport Schiphol, one of the three cargo terminals closes during the night while others remain operational, leading to inefficiencies in the handling of express shipments (XPS). During this period, XPS packages are temporarily stored at Terminal 3 (T3) instead of their typical location in Terminal 1 (T1), resulting in delays, missed flights, and disruptions to general cargo operations. This study analyzes and redesigns the night-time XPS handling process using process mapping, data analysis, and discrete event modeling and simulation (DES). A new system is proposed in which automated drop-off machines allow eligible XPS packages to be directly inserted into the T1 sorting system during the night, as opposed to the current system in which all packages must wait in a buffer until morning. Simulation models of both the current and proposed systems were developed using real operational data, and multiple routing scenarios were tested to evaluate the impact of varying proportions of directly accepted packages. The results indicate that the proposed system can significantly reduce package flow times and improve operational efficiency, with even a modest proportion of direct acceptance sufficient to outperform the current process. These findings demonstrate the potential of the proposed system to improve reliability, reduce congestion in T3, and enhance overall cargo handling performance, supporting further development and implementation.

1 Introduction

The air cargo industry plays a critical role in global logistics by enabling the rapid transport of time-sensitive goods. Efficient handling of express shipments is essential, as delays directly impact reliability, operational costs, and customer satisfaction [1]. At KLM Cargo's facilities at Amsterdam Airport Schiphol, operations are distributed across three cargo terminals, each with distinct functions. However, Terminal 1 (T1) [2], which handles express shipments (XPS), is closed during the night while the other terminals remain operational. As a result, night-time XPS deliveries are temporarily handled at Terminal 3 (T3) [3], leading to delays, congestion, and inefficiencies in both express and general cargo operations.

Despite extensive research on air cargo terminal operations, including process optimization [4] [5] [6], facility layout design [7] [8], time-dependent operations [9] [10], and the application of automation and artificial intelligence [11] [12] [13] [8], existing studies largely focus on fully operational or uniformly scheduled systems. There is a clear lack of research addressing environments in which cargo terminals operate under hybrid conditions, where some terminals temporarily close while others remain active. Such configurations introduce fluctuating capacity levels, routing challenges, and dynamic resource requirements that are not explored in depth in current literature.

This study addresses this gap by analyzing and redesigning the night-time handling process of express shipments at KLM Cargo. The current process is mapped and evaluated using document and data analysis, stakeholder input, on-site observations, interviews, modeling, and simulation. A new concept is then developed in which automated drop-off machines allow a portion of XPS packages to be directly processed and inserted into the sorting system during the night, while the remaining packages are routed for additional checks. The performance of both the current and proposed systems is evaluated using Discrete Event Simulation (DES), enabling a quantitative comparison under varying operational scenarios.

The main research question addressed in this study is:

How can the delivery process of express packages during nightly closures of T1 be redesigned to optimize flow, improve operational reliability, and minimize transport time and distance, while complying with relevant safety and stakeholder requirements?

To support this, six subquestions are also presented:

1. What are the size, layout, and capacity constraints of the physical spaces for the current processes and the proposed design and what are the quantities, types, physical characteristics, and storage patterns of cargo being delivered?
2. What are the sequential steps of and bottlenecks present in the current night-time delivery process, and which KPIs are appropriate to best evaluate and compare both the current and proposed processes?
3. What regulatory, safety, security, and stakeholder requirements must the new delivery system meet, and what relevant technologies, systems, or operational practices from other companies or industries could be incorporated into the final design?
4. How can the nightly cargo delivery process be designed efficiently with the information obtained through the previous sub-questions taken into account in order to achieve the highest possible improvements?
5. How can the proposed new design for the nightly cargo delivery process be best represented theoretically as a model?
6. How can this theoretical model be simulated and used to evaluate and compare the performance of the proposed design against the current state?

Answering these subquestions will provide a research-backed foundation for developing, evaluating, and validating an improved night-time delivery process for express shipments at KLM Cargo.

2 Methodology

The study was carried out in two main stages: analysis of the current process and development and evaluation of a solution design comprising a new process. First, the current night-time handling process for express shipments was examined using internal KLM Cargo documents, operational datasets, interviews with relevant personnel, and corroborated with on-site observations. These methods were used to map the existing process, identify bottlenecks, define system requirements, and establish the key performance indicators (KPIs) used throughout the study.

Internal documents, including handling manuals [14] [15], local work instructions [16], product portfolio documents [17], and terminal blueprints, were analyzed to understand current procedures, cargo characteristics, and spatial constraints. In addition, operational datasets [18] provided by KLM Cargo were processed to obtain input parameters for the simulation models, including truck interarrival times, truck waiting times, package counts, and package weight and volume distributions. Nine semi-structured interviews with staff from departments including Business Process Improvement (BPI), warehouse operations, security, and customs were conducted to obtain practical insights into current bottlenecks, requirements, and desired improvements. These insights were complemented by daytime and night-time in-person observations onsite, which were used to visually understand the current process and verify how operations were carried out in practice.

Based on the findings from the current-state analysis, a redesigned night-time delivery concept was developed through iterative brainstorming sessions. The proposed design consists of automated drop-off machines connected directly to the sorting system in Terminal 1 (T1), allowing eligible express packages to bypass the current overnight buffer at Terminal 3 (T3). Packages requiring further inspection are routed to a temporary holding area for additional checks which are to be sent into the sorter when T1 reopens.

To evaluate the performance of the proposed design, Discrete Event Simulation (DES) models were developed for both the current and redesigned processes. The models were implemented in Python and used empirical input data derived from KLM Cargo operations. Performance was assessed using both static and dynamic KPIs, including package flow time, truck waiting time, queue lengths, and the number of packages inserted into the sorter before T1 reopens. Multiple simulation runs were performed for each scenario in order to compare the operational performance of the two systems under varying routing conditions.

3 Current and new processes

3.1 Current process

KLM Cargo operates three cargo terminals at Amsterdam Airport Schiphol. Terminal 1 (T1) handles express shipments (XPS), while Terminals 2 and 3 primarily handle standard cargo flows. During the night, T1 is closed, but trucks carrying XPS shipments continue to arrive. As a result, these deliveries are redirected to Terminal 3 (T3), where they are temporarily accepted and stored. Once T1 reopens in the morning, the accumulated XPS cargo is transported from T3 back to T1, where it is manually inserted into the sorting system. This setup creates delays in the handling of express shipments, increases truck waiting times and congestion at T3, and causes XPS cargo to interfere with the general cargo operations carried out there.

3.2 Proposed new process

The proposed process eliminates the need for night-time storage of XPS cargo at T3 by allowing eligible shipments to be handled directly at T1 during the night. The new design consists of automated drop-off machines installed at T1 that are connected to scanning technology and the existing sorting system. Trucks arriving with XPS cargo unload their packages at these machines, after which packages are scanned and automatically routed. Packages that meet all operational and security requirements are inserted directly into the sorter, while packages requiring additional inspection are routed to temporary holding bins for manual checking the following morning. Packages that are too early, damaged, or require special handling are removed from the flow and do not use this process. In this way, the proposed system reduces the dependence on T3 and lessens the impact XPS packages have on its operations, shortens package flow times, and allows a portion of the XPS stream to be processed before T1 officially reopens.

4 Model and simulation

To compare the current and proposed night-time handling processes, Discrete Event Simulation (DES) was used. DES was selected as the system consists of time-dependent, sequential events such as truck arrivals, unloading, package scanning, buffering, and insertion into the sorter, all of which change the state of the system at discrete points in time. Two DES models were developed with one representing the current process and one representing the proposed process.

The models were parameterized using operational data provided by KLM Cargo covering June 2025 to January 2026. These data included truck interarrival times, truck waiting times, number of packages per truck, and package weight, and volume distributions. Several data cleaning and simplification steps were applied, including the exclusion of weekend arrivals, removal of invalid weight and volume records, and the assumption that weight and volume are evenly distributed across packages within the same shipment. Additional assumptions were required in order to represent the real-world process in a tractable model. These included fixed night-shift duration, constant unloading and scanning times, fixed resource capacities, and assumed package routing percentages. For both the current and proposed processes, 90% of arriving packages were assumed to be ultimately acceptable for sorting, while the remaining 10% were assumed to leave the modeled flow due to rejection, special handling requirements, or premature arrival.

For the 90% acceptable XPS packages, this was split between a percentage of packages that are accepted directly versus requires additional check. As this percentage split is not known with available data, multiple different splits were tested, starting with a perfect scenario of 90%/0% (in which all of the 90% are directly acceptable) and decreasing by factors of 10 until the most unoptimal split is reached (80/10, 70/20, etc.). This allows for a complete picture to be created of what the potential system can potentially produce and can be compared to the current process results in order to determine the threshold that will need to be achieved for the new proposal to be advantageous to implement.

The simulation models were implemented in Python using SimPy. Trucks and packages were represented as entities moving through a system of constrained resources, including docks, drop-off machines, handlers, and buffers. The current process model captures the arrival of XPS trucks at T3, unloading, package acceptance checks, temporary storage, and morning transfer to the sorter after T1 reopens. The proposed model captures unloading at automated drop-off machines, package scanning, routing into direct or additional-check flows, and morning processing of buffered packages. Multiple simulation runs were performed for each scenario, and key performance indicators such as package flow time, truck waiting time, queue lengths, and sorter insertion times were recorded. The validity of the model was assessed by comparing simulated arrival patterns and throughput levels with the corresponding historical KLM Cargo data.

5 Results

The simulation results were compared using the average mean XPS package flow time, defined as the time between a truck's arrival at the terminal and the moment the corresponding package is inserted into the sorter. Table 1 summarizes the mean flow time for the current process and for the ten routing scenarios of the proposed process, together with the absolute and relative difference compared with the current system. Figure 1 visualizes these results across the full range of direct/additional-check routing ratios.

Table 1: *Comparison of average mean XPS flow time between the current process and the new process routing scenarios*

Scenario	Mean flow time (min)	Δ Flow time (min)	Δ Flow time (%)
Current process	248.99	–	–
90/0	46.66	-202.33	-81.26%
80/10	66.55	-182.44	-73.27%
70/20	83.19	-165.80	-66.59%
60/30	115.35	-133.64	-53.67%
50/40	139.87	-109.12	-43.82%
40/50	176.69	-72.30	-29.04%
30/60	187.12	-61.87	-24.85%
20/70	232.57	-16.42	-6.59%
10/80	267.57	+18.58	+7.46%
0/90	283.61	+34.62	+13.90%

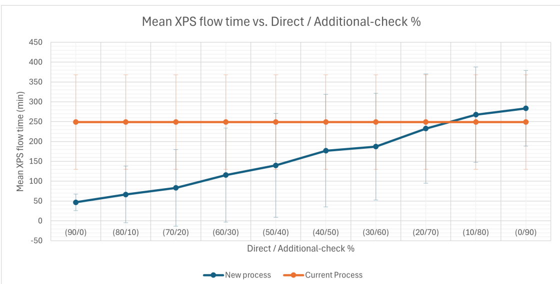


Figure 1: *Average mean XPS flow time for the current process and the evaluated direct/additional-check routing scenarios*

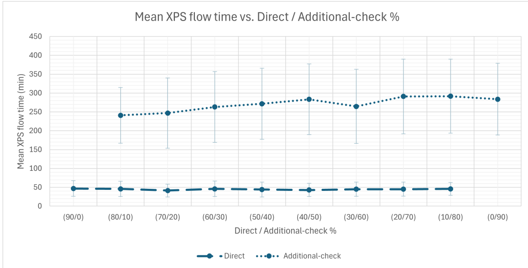


Figure 2: *Average mean XPS flow time for direct vs. additional-check flows over various direct/additional-check percentages*

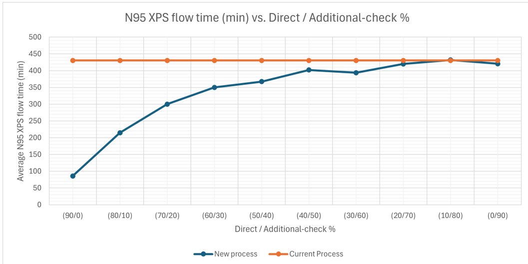


Figure 3: *Average N95 XPS flow time for both current process and various direct/additional-check percentages*

The results show a clear relationship between the percentage of packages that are directly accepted during the night and the resulting average package flow time. Lower average flow times are observed for scenarios with higher direct acceptance percentages, while scenarios with larger additional-check flow percentages produce progressively higher average flow times. Across the scenarios, the proposed process outperforms the current process for most routing configurations (above 15% directly acceptable).

6 Discussion

This section interprets the results of the simulation and compares the performance of the current and proposed processes.

6.1 Static process improvements

The proposed design significantly reduces transport distance for XPS packages from approximately 570 m to 10 m, representing a reduction of over 98%. In addition, the removal of XPS flows from T3 eliminates the need for approximately 120 m^2 of buffer space, while slightly increasing the required footprint in T1 from 75 m^2 to 140 m^2.

Overall, the redesign shifts operations closer to the sorter, reducing unnecessary transport and improving space utilization by eliminating inefficient use of pallet space for small shipments.

6.2 Simulation insights

The simulation results show a clear relationship between system performance and the proportion of directly accepted XPS packages. As the share of direct acceptance increases, the average package flow time decreases significantly, while high-percentile delays (N95) are reduced even more strongly.

In the current process, all acceptable packages are buffered until T1 reopens, leading to consistently high flow times (approximately 249 minutes). In contrast, the new process introduces a direct flow that allows immediate sorting, drastically reducing time in the system for those packages. Direct-flow packages exhibit low and consistent flow times (around 45 minutes), while additional-check packages behave similarly to the current process due to buffering.

6.3 Performance comparison

The results indicate that the proposed process outperforms the current process for most routing scenarios. A threshold is observed at approximately 15% direct acceptance, above which the new system achieves lower average flow times than the current process.

The primary driver of improvement is the ability to bypass buffering and insert packages directly into the sorter. This not only reduces average flow time but also improves reliability by reducing variability and extreme delays.

Although total process completion times remain similar across scenarios due to morning operations, the new process enables a substantial portion of packages to be sorted before T1 reopens, which is not possible in the current system.

6.4 Operational implications

From an operational perspective, the proposed system reduces congestion at T3, lowers truck waiting times, and decreases handling workload. Earlier sorting of packages improves delivery reliability and reduces the risk of missed flights. Additionally, minimizing transfers and intermediate storage reduces the likelihood of misplaced shipments.

6.5 Limitations

The model relies on several simplifying assumptions, including constant processing times, stable arrival patterns, and perfect system operation. Furthermore, key data such as the true distribution of package types and causes of missed departures were unavailable, requiring assumptions in the model.

While these limitations affect absolute accuracy, they do not compromise the validity of the relative comparison between the current and proposed processes, which remains the primary objective of this study.

7 Conclusion and recommendations

This study examined how the night-time handling process of express shipments at KLM

Cargo can be improved during the nightly closure of Terminal 1. A redesigned process was developed in which automated drop-off machines at T1 allow eligible XPS packages to be accepted and inserted directly into the sorter during the night, while other packages are routed to a buffer for additional checks. Discrete event simulation was used to compare the current and proposed processes under multiple routing scenarios. The results show that the proposed system can substantially reduce package flow times, decrease dependence on Terminal 3, and improve operational efficiency. In particular, the new process outperforms the current system when at least approximately 15% of packages can be directly accepted into the sorter. Since the modeled current process represents an idealized version without interference from general cargo trucks at T3, the practical benefits of the proposed system are likely to be even greater in reality.

Based on these results, it is recommended that KLM Cargo continue investigating the implementation of an automated drop-off system for night-time XPS deliveries. Further research should focus on obtaining more detailed data on package classifications, validating the automated compliance and routing logic, and evaluating system behavior under equipment failures or fluctuating demand conditions. In addition, the proposed concept may be relevant to other cargo or logistics facilities that operate under partial terminal closures or similar night-time handling constraints.

References

1. Grand Aire. The History of Air Cargo. 2021. URL: https://www.grandaire.com/news/the-history-of-air-cargo/.
2. KLM Real Estate & Facility Contracting. Plattegrond Begane Grond – Gebouw 545 Vrachtgebouw 1. Project: Vastgoed, Veiligheid en Milieu; Object: Gebouw 545 Vrachtgebouw 1; Drawing number: 11031; Internal engineering drawing. 1997.
3. KLM Real Estate & Facility Contracting. Plattegrond Begane Grond Laag-A – Vrachtgebouw 3 Schiphol Centrum. Project: Bordessen en roostervloer; Object: Vrachtgebouw 3 Schiphol Centrum; Drawing number: B26920P0; Internal engineering drawing. 2014.
4. Nath S P and Upadhyay RK. Reformation and optimization of cargo handling operation at Indian air cargo terminals. Journal of the Air Transport Research Society 2024;2:100022.
5. Lapphiphitmongkhon D. Improving air cargo handling operation: the case of an international airlines in Taiwan. 2008.
6. Paethrangsi N, Jaktong C, and Brahmaputra P. Determinants Affecting Air Cargo Operational Performance in Thailand: A Strategic and Mixed-Methods Approach. Interdisciplinary Academic and Research Journal 2025;5:1023–42.
7. Priansyah D, Ariyaka S, Syach GN, and Fasya YR. Planning of Cargo Terminal Area Requirement of Komodo Labuan Bajo International Airport. Advances in Transportation and Logistics Research 2024;7:839–53.
8. Ballis A. Overview of air cargo terminal design aspects. Transportation research record 2007;2007:117–25.
9. Gurturk M and Celik E. A Bibliometric Analysis of Air Cargo: Milestones, Trends, and Future Directions. Transportation Journal 2025;64:e70001.
10. Thein HW. A Study on Cargo Operation Management in Sky Cargo Terminal (Han Win Thein, 2025). PhD thesis. MERAL Portal, 2025.

11. Tubis AA, Jodejko-Pietruczuk A, and Nowakowski T. Use of augmented reality as a tool to support cargo handling operations at the CARGO air terminal. Sensors 2024;24:1099.

12. Hong SJ, Kim W, and Hiatt B. Examining airport agility at air cargo hub airports. Journal of Air Transport Management 2025;122:102710.

13. Chao CC and Yu PC. Quantitative evaluation model of air cargo competitiveness and comparative analysis of major Asia-Pacific airports. Transport Policy 2013;30:318–26.

14. KLM Cargo. Content, Terminology Training. In: *Cargo Handling Manual (CHM)*. Issued by SPL/FQ, internal operational manual. 2025. Chap. 1.

15. KLM Cargo. Accept Shipments as Ready-for-Carriage. In: *Cargo Handling Manual (CHM)*. Issued by SPL/FQ, internal operational manual. 2025. Chap. 8.

16. Talahatu B. Oogplein afhandeling. Operationele Instructies. Version 3. Validated by Shiftleader EPS, internal operational document. KLM Cargo. 2021.

17. Air France KLM Martinair Cargo. Product Portfolio 2025. Company product overview document describing cargo services and requirements, including express and express heavy shipments. Air France KLM Martinair Cargo, 2025. URL: `https://www.afklcargo.com/MY/en/products_and_solutions/products_and_solutions` (visited on 03/04/2026).

18. KLM Cargo. Operational datasets for cargo handling and truck arrivals. Internal operational datasets provided by KLM Cargo for thesis research. Includes truck arrival times, docking times, waiting times, and express shipment characteristics. 2025.

B

Appendix B

B.1. Declaration of usage of AI

The usage of AI-assisted tools such as ChatGPT and Claude was performed in order to:

- help with Excel and how to use its features to process raw data to obtain the results I wanted
- help with Python and Visual Studio Code
- help answer questions about Discrete Event Simulation
- help with eliminating repetition in my writing by suggesting synonyms
- help with Latex and Overleaf as I typically use Microsoft Word and so formatting and structuring in Latex is challenging for me and this helps save time and lets me focus on the actual content of the report instead of spending inordinate amounts of time on figuring out how to create a text box or make text a specific color

Appendix C

C.1. Simulation code

C.1.1. entities.py

```python
# ============================================================
# entities.py
# Entity definitions for the simulation models
# ============================================================

class Truck:
    """
    Represents a truck arriving at the cargo terminal.

    Attributes:
        id (int): Unique truck identifier.
        num_packages (int): Number of XPS packages carried by the truck.
        arrival_time_CURRENT (float | None): Arrival time in the current process.
        arrival_time_NEW (float | None): Arrival time in the new process.
    """

    def __init__(self, truck_id, num_packages):
        self.id = truck_id
        self.num_packages = num_packages
        self.arrival_time_CURRENT = None
        self.arrival_time_NEW = None


class XPSPackage:
    """
    Represents a single XPS package handled in the simulation.

    Attributes:
        truck_id (int): ID of the truck the package arrived with.
        package_id (int): Unique package identifier within the truck.
        package_type (str): Package category, for example:
            'accepted', 'SHC', 'rejected', or any new-process category.
        arrival_time_CURRENT (float | None): Time the package enters the current
            process.
        end_time_CURRENT (float | None): Time the package finishes the current
            process.
        arrival_time_NEW (float | None): Time the package enters the new process.
        end_time_NEW (float | None): Time the package finishes the new process.
    """
```

```python

    def __init__(self, truck_id, package_id, package_type="accepted"):
        self.truck_id = truck_id
        self.package_id = package_id
        self.package_type = package_type
        self.arrival_time_CURRENT = None
        self.end_time_CURRENT = None
        self.arrival_time_NEW = None
        self.end_time_NEW = None
```

C.1.2. resources.py

```python
# ============================================================
# resources.py
# Simulation resource definitions for current and new processes
# ============================================================

import simpy

from parameters import (
    NUMBER_OF_DOCKS_CURRENT,
    T3_HANDLER_COUNT_CURRENT,
    T1_HANDLER_COUNT_CURRENT,
    NUMBER_OF_DROP_OFF_MACHINES_NEW,
    T1_HANDLER_COUNT_NEW,
)


def create_resources(env):
    """
    Create all SimPy resources used in the simulation models.

    Parameters
    ----------
    env : simpy.Environment
        The simulation environment.

    Returns
    -------
    dict
        Dictionary containing the resources for both the current and new
        process designs.
    """
    return {
        # Current process
        "dock_T3_CURRENT": simpy.Resource(env, capacity=NUMBER_OF_DOCKS_CURRENT),
        "T3_handler_CURRENT": simpy.Resource(env, capacity=
            T3_HANDLER_COUNT_CURRENT),
        "T1_handler_CURRENT": simpy.Resource(env, capacity=
            T1_HANDLER_COUNT_CURRENT),

        # New process
        "drop_machines_NEW": simpy.Resource(
            env, capacity=NUMBER_OF_DROP_OFF_MACHINES_NEW
        ),
        "T1_handler_NEW": simpy.Resource(env, capacity=T1_HANDLER_COUNT_NEW),
    }
```

C.1.3. parameters.py

```python
# ============================================================
# parameters.py
```

```
# Simulation parameters for the CURRENT and NEW XPS processes
# Time unit: minutes
# ============================================================

# ----------------------------
# Random seed
# ----------------------------
SEED = 43
# Seed used for reproducibility of random sampling

NUM_ITERATIONS = 100
# Number of iterations that the code will go through

# ============================================================
# CURRENT PROCESS PARAMETERS (_CURRENT)
# ============================================================

# ----------------------------
# Simulation time
# ----------------------------
SIMULATION_END_CURRENT = 480
# Total simulation duration for the current process
# 480 minutes = 8 hours

# ----------------------------
# Truck input parameters
# ----------------------------
TRUCKS_LAMBDA_CURRENT = 20
# Legacy parameter name from earlier model version.
# Currently no longer used for truck generation.

PACKAGES_LAMBDA_CURRENT = 10
# Legacy parameter name from earlier model version.
# Currently no longer used for package generation.

# ----------------------------
# Dock / unloading (T3 side)
# ----------------------------
UNLOADING_TIME_CURRENT = 20
# Time required to unload one truck at T3

NUMBER_OF_DOCKS_CURRENT = 2
# Number of unloading docks available at T3

# ----------------------------
# T3 scanning
# ----------------------------
T3_HANDLER_COUNT_CURRENT = 2
# Number of human checkers available to scan XPS packages at T3

T3_SCAN_TIME_CURRENT = 3
# Time required to scan one XPS package at T3

# ----------------------------
# XPS routing percentages
# ----------------------------
XPS_ACCEPTED_CURRENT = 0.90
# Share of packages accepted and sent to belly-cart buffer

XPS_SHC_CURRENT = 0.08
# Share of packages requiring special handling
```

```

XPS_REJECTED_CURRENT = 0.02
# Share of packages rejected from the process

# ----------------------------
# T1 sorter processing
# ----------------------------
T1_HANDLER_COUNT_CURRENT = 2
# Number of handlers available to insert packages into the sorter at T1

T1_SORT_TIME_CURRENT = 0.25
# Time required to insert one package into the sorter at T1


# ============================================================
# NEW PROCESS PARAMETERS (_NEW)
# ============================================================

# ----------------------------
# Simulation time
# ----------------------------
SIMULATION_END_NEW = 480
# Total simulation duration for the new process
# 480 minutes = 8 hours

# ----------------------------
# Truck input parameters
# ----------------------------
TRUCK_INTERARRIVAL_TIME_NEW = 8
# Legacy parameter name from earlier model version.
# Currently no longer used for truck generation.

PACKAGES_PER_TRUCK_NEW = 7
# Legacy parameter name from earlier model version.
# Currently no longer used for package generation.

# ----------------------------
# Drop-off / unloading (T1 side)
# ----------------------------
UNLOADING_TIME_NEW = 20
# Time required to unload one truck at the drop-off machine

NUMBER_OF_DROP_OFF_MACHINES_NEW = 2
# Number of drop-off machines available in the new process

# ----------------------------
# Automated scanning
# ----------------------------
AUTO_SCAN_TIME_NEW = 3
# Time required for one package to be scanned by the drop-off machine

# ----------------------------
# XPS routing percentages
# ----------------------------
NEW_ACCEPTED_DIRECT_NEW = 0.60
# Share of packages directly accepted and sent into the sorter

NEW_ADDITIONAL_CHECK_NEW = 0.30
# Share of packages requiring additional human check

NEW_TOO_EARLY_NEW = 0.05
```

```
# Share of packages arriving too early and removed from the system

NEW_REJECTED_TO_T3_NEW = 0.05
# Share of packages rejected and sent to T3

# ---------------------------
# Manual additional-check processing
# ---------------------------
RECHECK_SCAN_TIME_NEW = 3
# Legacy parameter name from earlier model version.
# Currently not used in the active model.

T1_HANDLER_COUNT_NEW = 2
# Number of handlers available to manually process additional-check packages

T1_SORT_TIME_NEW = 3
# Time required for a handler to insert one additional-check package into the
    sorter
```

C.1.4. kpi_tracking.py

```
# ============================================================
# kpi_tracking.py
# KPI initialization and printing for CURRENT and NEW processes
# ============================================================

import numpy as np
from parameters import SIMULATION_END_CURRENT, SIMULATION_END_NEW


# ============================================================
# Initialize KPI dictionary (CURRENT)
# ============================================================
def initialize_kpis():
    kpi = {
        # ----------------------------
        # Time triggers
        # ----------------------------
        "t_reopen": SIMULATION_END_CURRENT,
        "t_lastunload": 0.0,
        "t_lastXPSscanned": 0.0,
        "t_totalprocess": 0.0,
        "t_firstXPSinserted": None,
        "t_lastXPSinserted": 0.0,

        # ----------------------------
        # Trucks
        # ----------------------------
        "T_arrived": 0,
        "T_beforeT1reopen": 0,
        "T_currentlydocked": 0,
        "T_currentlydocked_at_reopen": 0,
        "T_waitinginqueue": 0,
        "T_waitinginqueue_at_reopen": 0,
        "T_processed": 0,
        "truck_waiting_times": [],
        "inter_arrival_times": [],

        # ----------------------------
        # Debug / input tracking
        # ----------------------------
```

```
        "truck_arrival_times": [],
        "truck_package_counts": [],

        # ----------------------------
        # Packages
        # ----------------------------
        "XPS_scanned_total": 0,
        "XPS_total": 0,
        "XPS_acceptable": 0,
        "XPS_SHC": 0,
        "XPS_unacceptable": 0,
        "XPS_flow_times": [],


        # ----------------------------
        # System
        # ----------------------------
        "peak_dock_queue_length": 0,
        "sorting_ready": False,
    }
    return kpi


# ============================================================
# Create summary KPI dictionary for ONE CURRENT-process run
# ============================================================
def summarize_kpis_current(kpi):
    """
    Converts one full KPI dictionary from one simulation run
    into a smaller summary dictionary with only numeric results.
    """
    avg_wait = np.mean(kpi["truck_waiting_times"]) if kpi["truck_waiting_times"] else 0.0
    n95_wait = np.percentile(kpi["truck_waiting_times"], 95) if kpi["truck_waiting_times"] else 0.0

    flow_times = [t for t in kpi.get("XPS_flow_times", []) if t > 0]
    if flow_times:
        max_flow = max(flow_times)
        min_flow = min(flow_times)
        avg_flow = np.mean(flow_times)
        median_flow = np.median(flow_times)
        std_flow = np.std(flow_times)
        n95_flow = np.percentile(flow_times, 95)
    else:
        max_flow = 0.0
        min_flow = 0.0
        avg_flow = 0.0
        median_flow = 0.0
        std_flow = 0.0
        n95_flow = 0.0

    first_inserted = kpi["t_firstXPSinserted"] if kpi["t_firstXPSinserted"] is not None else 0.0

    summary = {
        "t_reopen": kpi["t_reopen"],
        "t_lastunload": kpi["t_lastunload"],
        "t_lastXPSscanned": kpi["t_lastXPSscanned"],
        "t_totalprocess": kpi["t_totalprocess"],
        "t_firstXPSinserted": first_inserted,
```

```
        "t_lastXPSinserted": kpi["t_lastXPSinserted"],

        "T_arrived": kpi["T_arrived"],
        "T_beforeT1reopen": kpi["T_beforeT1reopen"],
        "T_currentlydocked_at_reopen": kpi["T_currentlydocked_at_reopen"],
        "T_waitinginqueue_at_reopen": kpi["T_waitinginqueue_at_reopen"],
        "T_processed": kpi["T_processed"],
        "peak_dock_queue_length": kpi["peak_dock_queue_length"],

        "XPS_scanned_total": kpi["XPS_scanned_total"],
        "XPS_acceptable": kpi["XPS_acceptable"],
        "XPS_SHC": kpi["XPS_SHC"],
        "XPS_unacceptable": kpi["XPS_unacceptable"],

        "avg_truck_waiting_time": avg_wait,
        "n95_truck_waiting_time": n95_wait,

        "max_xps_flow_time": max_flow,
        "min_xps_flow_time": min_flow,
        "avg_xps_flow_time": avg_flow,
        "median_xps_flow_time": median_flow,
        "std_xps_flow_time": std_flow,
        "n95_xps_flow_time": n95_flow,
    }

    return summary


# ============================================================
# Average summary dictionaries over multiple CURRENT runs
# ============================================================
def average_kpi_summaries_current(all_summaries):
    """
    Averages a list of summary dictionaries produced by
    summarize_kpis_current().
    """
    if not all_summaries:
        return {}

    averaged = {}
    keys = all_summaries[0].keys()

    for key in keys:
        averaged[key] = float(np.mean([summary[key] for summary in all_summaries])
            )

    return averaged


# ============================================================
# Print KPI results (CURRENT) for ONE run
# ============================================================
def print_kpis_current(kpi):
    print("\n======================================================================"
        )
    print("                 CURRENT PROCESS SIMULATION RESULTS")
    print("======================================================================\n"
        )

    # ------------------------------------------------
    # Debug print of generated arrivals
```

```
    # ------------------------------------------------
    num_trucks = len(kpi.get("truck_arrival_times", []))
    interarrival_times = [int(x) for x in kpi.get("inter_arrival_times", [])]
    arrival_times = [int(x) for x in kpi.get("truck_arrival_times", [])]
    packages = kpi.get("truck_package_counts", [])

    print("GENERATED TRUCK INPUT DATA:")

    print(f"{'Number of trucks:':20}{num_trucks}")

    inter_str = " ".join(f"{x:4d}" for x in interarrival_times)
    arr_str = " ".join(f"{x:4d}" for x in arrival_times)
    pkg_str = " ".join(f"{x:4d}" for x in packages)

    print(f"{'Interarrival times (min):':20}{inter_str}")
    print(f"{'Arrival times (min):     ':20}{arr_str}")
    print(f"{'Packages per truck:      ':20}{pkg_str}")

    print()

    # 1) t_reopen
    print("-----------------------------------------------------------------------------------")
    print("1) t_reopen  (Night shift ends):  " f"{kpi['t_reopen']:.2f} min")
    print("-----------------------------------------------------------------------------------")
    print(f"Total Trucks arrived during night shift     (T_arrived)            : {kpi['T_arrived']}")
    print(f"Total Trucks processed before T1 reopens    (T_beforeT1reopen)     : {kpi['T_beforeT1reopen']} / {kpi['T_arrived']}")
    print(f"Total Trucks still docked and unloading     (T_currentlydocked)    : {kpi['T_currentlydocked_at_reopen']} / {kpi['T_arrived']}")
    print(f"Total Trucks still waiting in queue         (T_waitinginqueue)     : {kpi['T_waitinginqueue_at_reopen']} / {kpi['T_arrived']}\n")

    # 2) t_lastunload
    avg_wait = np.mean(kpi["truck_waiting_times"]) if kpi["truck_waiting_times"] else 0
    n95_wait = np.percentile(kpi["truck_waiting_times"], 95) if kpi["truck_waiting_times"] else 0

    print("-----------------------------------------------------------------------------------")
    print("2) t_lastunload  (Last truck finished unloading):  " f"{kpi['t_lastunload']:.2f} min")
    print("-----------------------------------------------------------------------------------")
    print(f"Total Trucks processed          (T_processed)   :  {kpi['T_processed']} trucks")
    print(f"Peak dock queue length:         (T_peak)        :  {kpi['peak_dock_queue_length']} trucks")
    print(f"Average truck waiting time:     (t_Tavg)        :  {avg_wait:.2f} min")
    print(f"N95 truck waiting time:         (t_TN95)        :  {n95_wait:.2f} min\n")

    # 3) t_lastXPSscanned
```

```
    print("--------------------------------------------------------------------------------------")
    print("3) t_lastXPSscanned  (Last XPS scanned):  " f"{kpi['t_lastXPSscanned']:.2f} min")
    print("--------------------------------------------------------------------------------------")
    print(f"Total XPS packages removed and scanned  (XPS_scanned_total) :  {kpi['XPS_scanned_total']}")
    print(f"Total XPS packages accepted             (XPS_acceptable)    :  {kpi['XPS_acceptable']} / {kpi['XPS_scanned_total']}")
    print(f"Total XPS packages requiring SHC        (XPS_SHC)           :  {kpi['XPS_SHC']} / {kpi['XPS_scanned_total']}")
    print(f"Total XPS packages not accepted         (XPS_unacceptable)  :  {kpi['XPS_unacceptable']} / {kpi['XPS_scanned_total']}\n")

    # 4) t_totalprocess
    flow_times = [t for t in kpi.get("XPS_flow_times", []) if t > 0]
    if flow_times:
        max_flow = max(flow_times)
        min_flow = min(flow_times)
        avg_flow = np.mean(flow_times)
        median_flow = np.median(flow_times)
        std_flow = np.std(flow_times)
        n95_flow = np.percentile(flow_times, 95)
    else:
        max_flow = min_flow = avg_flow = median_flow = std_flow = n95_flow = 0.0

    print("--------------------------------------------------------------------------------------")
    print("4) t_totalprocess  (Last XPS sorted):  " f"{kpi['t_totalprocess']:.2f} min")
    print("--------------------------------------------------------------------------------------")

    print(f"Total XPS packages inserted into sorter   (XPS_sorter)   :  {kpi['XPS_acceptable']}\n")

    first_inserted = kpi["t_firstXPSinserted"] if kpi["t_firstXPSinserted"] is not None else 0.0

    print(f"Time first XPS enters sorter      (t_firstXPSinserted)   :  {first_inserted:.2f} min")
    print(f"Time last XPS enters sorter       (t_lastXPSinserted)    :  {kpi['t_lastXPSinserted']:.2f} min\n")

    print(f"Max time an XPS package spent in system     (t_XPSmax)   :  {max_flow:.2f} min")
    print(f"Min time an XPS package spent in system     (t_XPSmin)   :  {min_flow:.2f} min")
    print(f"Average time XPS package in system          (t_XPSavg)   :  {avg_flow:.2f} min")
    print(f"Median time XPS package in system           (t_XPSmed)   :  {median_flow:.2f} min")
    print(f"Std. dev. XPS package in system             (t_XPSstd)   :  {std_flow:.2f} min")
    print(f"N95 XPS package in system                   (t_XPSN95)   :  {n95_flow:.2f} min\n")
```

```
# ============================================================
# Create across-run input statistics (CURRENT)
# ============================================================
def summarize_input_statistics(values):
    """
    Creates summary statistics across simulation runs for a list of values,
    for example trucks per run or packages per run.
    """
    summary = {
        "max": float(np.max(values)),
        "min": float(np.min(values)),
        "median": float(np.median(values)),
        "mean": float(np.mean(values)),
        "std": float(np.std(values)),
        "max_run": int(np.argmax(values)) + 1,
        "min_run": int(np.argmin(values)) + 1,
    }
    return summary


# ============================================================
# Print across-run input statistics (CURRENT)
# ============================================================
def print_input_statistics_current(truck_stats, package_stats, num_iterations):
    print("\n======================================================================")
    print(f"      INPUT STATISTICS ACROSS CURRENT PROCESS RUNS ({num_iterations} RUNS)")
    print("======================================================================\n")

    print("TRUCKS PER RUN")
    print(f"Maximum trucks in one run        :  {truck_stats['max']:.0f} (Run {truck_stats['max_run']})")
    print(f"Minimum trucks in one run        :  {truck_stats['min']:.0f} (Run {truck_stats['min_run']})")
    print(f"Median trucks per run            :  {truck_stats['median']:.2f}")
    print(f"Average trucks per run           :  {truck_stats['mean']:.2f}")
    print(f"Std. dev. trucks per run         :  {truck_stats['std']:.2f}\n")

    print("PACKAGES PER RUN")
    print(f"Maximum packages in one run      :  {package_stats['max']:.0f} (Run {package_stats['max_run']})")
    print(f"Minimum packages in one run      :  {package_stats['min']:.0f} (Run {package_stats['min_run']})")
    print(f"Median packages per run          :  {package_stats['median']:.2f}")
    print(f"Average packages per run         :  {package_stats['mean']:.2f}")
    print(f"Std. dev. packages per run       :  {package_stats['std']:.2f}")



# ============================================================
# Print averaged KPI results (CURRENT) over multiple runs
# ============================================================
def print_averaged_kpis_current(avg_kpi, num_iterations):
    print("\n======================================================================"
```

```
    )
    print(f"      AVERAGED CURRENT PROCESS RESULTS OVER {num_iterations} RUNS")
    print("======================================================================\n")

    print(f"Average trucks arrived                     :  {avg_kpi['T_arrived']:.2f}")
    print(f"Average trucks processed before T1 reopen  :  {avg_kpi['T_beforeT1reopen']:.2f}")
    print(f"Average trucks processed total             :  {avg_kpi['T_processed']:.2f}")
    print(f"Average trucks waiting at reopen           :  {avg_kpi['T_waitinginqueue_at_reopen']:.2f}")
    print(f"Average trucks docked at reopen            :  {avg_kpi['T_currentlydocked_at_reopen']:.2f}\n")

    print(f"Average last unload time                   :  {avg_kpi['t_lastunload']:.2f} min")
    print(f"Average last XPS scanned time              :  {avg_kpi['t_lastXPSscanned']:.2f} min")
    print(f"Average first XPS inserted into sorter     :  {avg_kpi['t_firstXPSinserted']:.2f} min")
    print(f"Average last XPS inserted into sorter      :  {avg_kpi['t_lastXPSinserted']:.2f} min")
    print(f"Average total process time                 :  {avg_kpi['t_totalprocess']:.2f} min\n")

    print(f"Average truck waiting time                 :  {avg_kpi['avg_truck_waiting_time']:.2f} min")
    print(f"Average truck waiting N95                  :  {avg_kpi['n95_truck_waiting_time']:.2f} min\n")

    print(f"Average XPS scanned total                  :  {avg_kpi['XPS_scanned_total']:.2f}")
    print(f"Average XPS accepted                       :  {avg_kpi['XPS_acceptable']:.2f}")
    print(f"Average XPS SHC                            :  {avg_kpi['XPS_SHC']:.2f}")
    print(f"Average XPS unacceptable                   :  {avg_kpi['XPS_unacceptable']:.2f}\n")

    print(f"Average mean XPS flow time                 :  {avg_kpi['avg_xps_flow_time']:.2f} min")
    print(f"Average median XPS flow time               :  {avg_kpi['median_xps_flow_time']:.2f} min")
    print(f"Average N95 XPS flow time                  :  {avg_kpi['n95_xps_flow_time']:.2f} min")
    print(f"Average min XPS flow time                  :  {avg_kpi['min_xps_flow_time']:.2f} min")
    print(f"Average max XPS flow time                  :  {avg_kpi['max_xps_flow_time']:.2f} min")
    print(f"Average std. dev. XPS flow time            :  {avg_kpi['std_xps_flow_time']:.2f} min\n")
```

```python
# ============================================================
# Initialize KPI dictionary (NEW)
# ============================================================
def initialize_kpis_new():
    kpi = {
        # ----------------------------
        # Time triggers (NEW process)
        # ----------------------------
        "t_reopen": SIMULATION_END_NEW,
        "t_lastunload": 0.0,
        "t_lastXPSscanned": 0.0,
        "t_lastdirectXPS": 0.0,
        "t_lastadditionalXPS": 0.0,
        "t_totalprocess": 0.0,

        "t_firstdirectXPS": None,
        "t_firstadditionalXPS": None,

        # ----------------------------
        # Trucks (NEW process)
        # ----------------------------
        "T_arrived": 0,
        "T_beforeT1reopen": 0,
        "T_currentlydocked": 0,
        "T_currentlydocked_at_reopen": 0,
        "T_waitinginqueue": 0,
        "T_waitinginqueue_at_reopen": 0,
        "T_processed": 0,
        "truck_waiting_times": [],

        # ----------------------------
        # Debug / input tracking
        # ----------------------------
        "truck_arrival_times": [],
        "truck_interarrival_times": [],
        "truck_package_counts": [],

        # ----------------------------
```

```
        # Packages (NEW process)
        # ---------------------------
        "XPS_spawned_total": 0,
        "XPS_scanned_total": 0,
        "XPS_total": 0,

        "XPS_acceptable": 0,
        "XPS_additional_check": 0,
        "XPS_too_early": 0,
        "XPS_rejected_to_T3": 0,
        "XPS_unacceptable": 0,
        "XPS_inserted_by_reopen": 0,
        "XPS_inserted_after_reopen": 0,



        # ---------------------------
        # Flow times (NEW process)
        # ---------------------------
        "XPS_flow_times": [],
        "XPS_flow_times_direct": [],
        "XPS_flow_times_additional": [],

        # ---------------------------
        # System (NEW process)
        # ---------------------------
        "peak_drop_queue_length": 0,

        # ---------------------------
        # Control (NEW process)
        # ---------------------------
        "arrivals_done": False,
    }
    return kpi


# ============================================================
# Create summary KPI dictionary for ONE NEW-process run
# ============================================================
def summarize_kpis_new(kpi):
    """
    Converts one full KPI dictionary from one NEW-process simulation run
    into a smaller summary dictionary with only numeric results.
    """
    avg_wait = np.mean(kpi["truck_waiting_times"]) if kpi["truck_waiting_times"]
        else 0.0
    n95_wait = np.percentile(kpi["truck_waiting_times"], 95) if kpi["
        truck_waiting_times"] else 0.0

    def _stats(arr):
        arr = [x for x in arr if x > 0]
        if not arr:
            return 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
        return (
            max(arr),
            min(arr),
            float(np.mean(arr)),
            float(np.median(arr)),
            float(np.std(arr)),
            float(np.percentile(arr, 95))
        )
```

```
    d_max, d_min, d_avg, d_median, d_std, d_n95 = _stats(kpi.get("XPS_flow_times_direct", []))
    a_max, a_min, a_avg, a_median, a_std, a_n95 = _stats(kpi.get("XPS_flow_times_additional", []))
    o_max, o_min, o_avg, o_median, o_std, o_n95 = _stats(kpi.get("XPS_flow_times", []))

    first_direct = kpi["t_firstdirectXPS"] if kpi["t_firstdirectXPS"] is not None else 0.0
    first_additional = kpi["t_firstadditionalXPS"] if kpi["t_firstadditionalXPS"] is not None else 0.0

    summary = {
        "t_reopen": kpi["t_reopen"],
        "t_lastunload": kpi["t_lastunload"],
        "t_lastXPSscanned": kpi["t_lastXPSscanned"],
        "t_firstdirectXPS": first_direct,
        "t_lastdirectXPS": kpi["t_lastdirectXPS"],
        "t_firstadditionalXPS": first_additional,
        "t_lastadditionalXPS": kpi["t_lastadditionalXPS"],
        "t_totalprocess": kpi["t_totalprocess"],

        "T_arrived": kpi["T_arrived"],
        "T_beforeT1reopen": kpi["T_beforeT1reopen"],
        "T_currentlydocked_at_reopen": kpi["T_currentlydocked_at_reopen"],
        "T_waitinginqueue_at_reopen": kpi["T_waitinginqueue_at_reopen"],
        "T_processed": kpi["T_processed"],
        "peak_drop_queue_length": kpi["peak_drop_queue_length"],

        "XPS_spawned_total": kpi["XPS_spawned_total"],
        "XPS_scanned_total": kpi["XPS_scanned_total"],
        "XPS_total": kpi["XPS_total"],
        "XPS_acceptable": kpi["XPS_acceptable"],
        "XPS_additional_check": kpi["XPS_additional_check"],
        "XPS_too_early": kpi["XPS_too_early"],
        "XPS_rejected_to_T3": kpi["XPS_rejected_to_T3"],
        "XPS_unacceptable": kpi["XPS_unacceptable"],
        "XPS_inserted_by_reopen": kpi["XPS_inserted_by_reopen"],
        "XPS_inserted_after_reopen": kpi["XPS_inserted_after_reopen"],


        "avg_truck_waiting_time": avg_wait,
        "n95_truck_waiting_time": n95_wait,

        "max_direct_flow_time": d_max,
        "min_direct_flow_time": d_min,
        "avg_direct_flow_time": d_avg,
        "median_direct_flow_time": d_median,
        "std_direct_flow_time": d_std,
        "n95_direct_flow_time": d_n95,

        "max_additional_flow_time": a_max,
        "min_additional_flow_time": a_min,
        "avg_additional_flow_time": a_avg,
        "median_additional_flow_time": a_median,
        "std_additional_flow_time": a_std,
        "n95_additional_flow_time": a_n95,

        "max_total_flow_time": o_max,
        "min_total_flow_time": o_min,
```

```
        "avg_total_flow_time": o_avg,
        "median_total_flow_time": o_median,
        "std_total_flow_time": o_std,
        "n95_total_flow_time": o_n95,
    }

    return summary


# ============================================================
# Average summary dictionaries over multiple NEW runs
# ============================================================
def average_kpi_summaries_new(all_summaries):
    """
    Averages a list of summary dictionaries produced by
    summarize_kpis_new().
    """
    if not all_summaries:
        return {}

    averaged = {}
    keys = all_summaries[0].keys()

    for key in keys:
        averaged[key] = float(np.mean([summary[key] for summary in all_summaries]))

    return averaged


# ============================================================
# Print KPI results (NEW)
# ============================================================
def print_kpis_new(kpi):
    print("\n======================================================================")
    print("                    NEW PROCESS SIMULATION RESULTS")
    print("======================================================================\n")

    # ------------------------------------------------
    # Debug print of generated arrivals (NEW process)
    # ------------------------------------------------
    num_trucks = len(kpi.get("truck_arrival_times", []))
    interarrival_times = [int(x) for x in kpi.get("truck_interarrival_times", [])]
    arrival_times = [int(x) for x in kpi.get("truck_arrival_times", [])]
    packages = kpi.get("truck_package_counts", [])

    print("GENERATED TRUCK INPUT DATA:")

    print(f"{'Number of trucks:':20}{num_trucks}")

    inter_str = " ".join(f"{x:4d}" for x in interarrival_times)
    arr_str = " ".join(f"{x:4d}" for x in arrival_times)
    pkg_str = " ".join(f"{x:4d}" for x in packages)

    print(f"{'Interarrival times:':20}{inter_str}")
    print(f"{'Arrival times:':20}{arr_str}")
    print(f"{'Packages per truck:':20}{pkg_str}")

    print()
```

```python

    # 1) t_reopen
    print("----------------------------------------------------------------------------------")
    print("1) t_reopen  (Night shift ends):  " f"{kpi['t_reopen']:.2f} min")
    print("----------------------------------------------------------------------------------")
    print(f"Total Trucks arrived during night shift     (T_arrived)           : {kpi['T_arrived']}")
    print(f"Total Trucks processed before T1 reopens    (T_beforeT1reopen)    : {kpi['T_beforeT1reopen']} / {kpi['T_arrived']}")
    print(f"Total Trucks still docked and unloading     (T_currentlydocked)   : {kpi['T_currentlydocked_at_reopen']} / {kpi['T_arrived']}")
    print(f"Total Trucks still waiting in queue         (T_waitinginqueue)    : {kpi['T_waitinginqueue_at_reopen']} / {kpi['T_arrived']}\n")

    # 2) t_lastunload
    avg_wait = np.mean(kpi["truck_waiting_times"]) if kpi["truck_waiting_times"] else 0
    n95_wait = np.percentile(kpi["truck_waiting_times"], 95) if kpi["truck_waiting_times"] else 0

    print("----------------------------------------------------------------------------------")
    print("2) t_lastunload  (Last truck finished unloading):  " f"{kpi['t_lastunload']:.2f} min")
    print("----------------------------------------------------------------------------------")
    print(f"Total Trucks processed              (T_processed)   :  {kpi['T_processed']} trucks")
    print(f"Peak drop-off machine queue length  (T_droppeak)    :  {kpi['peak_drop_queue_length']} trucks")
    print(f"Average truck waiting time          (t_Tavg)        :  {avg_wait:.2f} min")
    print(f"N95 truck waiting time              (t_TN95)        :  {n95_wait:.2f} min\n")

    # 3) scanning totals + routing
    print("----------------------------------------------------------------------------------")
    print(f"3) t_lastXPSscanned (Last auto-scan complete, any route) :  {kpi['t_lastXPSscanned']:.2f} min")
    print("----------------------------------------------------------------------------------")

    print(f"XPS scanned total                       (XPS_total)              :  {kpi['XPS_scanned_total']}")
    print(f"Accepted direct                         (XPS_accepteddirect)     :  {kpi['XPS_acceptable']} / {kpi['XPS_scanned_total']}")
    print(f"Additional check (buffered)             (XPS_additionalcheck)    :  {kpi['XPS_additional_check']} / {kpi['XPS_scanned_total']}")
    print(f"Too early                               (XPS_tooearly)           :  {kpi['XPS_too_early']} / {kpi['XPS_scanned_total']}")
    print(f"Rejected and sent to T3                 (XPS_rejected)           :  {kpi['XPS_rejected_to_T3']} / {kpi['XPS_scanned_total']}\n")
```

```
    # 4) Completion times + per-stream flow time stats
    def _stats(arr):
        arr = [x for x in arr if x > 0]
        if not arr:
            return (0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
        return (
            max(arr),
            min(arr),
            float(np.mean(arr)),
            float(np.median(arr)),
            float(np.std(arr)),
            float(np.percentile(arr, 95))
        )

    d_max, d_min, d_avg, d_median, d_std, d_n95 = _stats(kpi.get("XPS_flow_times_direct", []))
    a_max, a_min, a_avg, a_median, a_std, a_n95 = _stats(kpi.get("XPS_flow_times_additional", []))
    o_max, o_min, o_avg, o_median, o_std, o_n95 = _stats(kpi.get("XPS_flow_times", []))

    print("------------------------------------------------------------------------------------------------")
    print("4) t_totalprocess (XPS inserted into sorter):  " f"{kpi['t_totalprocess']:.2f} min")
    print("------------------------------------------------------------------------------------------------")

    total_sorter = kpi["XPS_acceptable"] + kpi["XPS_additional_check"]

    print(f"Total XPS packages inserted into sorter (XPS_sorter)       :  {total_sorter} / {kpi['XPS_scanned_total']}\n")

    print(f"XPS inserted into sorter by t = 480      (XPS_by_reopen)    :  {kpi['XPS_inserted_by_reopen']}")
    print(f"XPS inserted into sorter after t = 480   (XPS_after_reopen):  {kpi['XPS_inserted_after_reopen']}\n")


    first_direct = kpi["t_firstdirectXPS"] if kpi["t_firstdirectXPS"] is not None else 0.0
    last_direct = kpi["t_lastdirectXPS"]
    first_additional = kpi["t_firstadditionalXPS"] if kpi["t_firstadditionalXPS"] is not None else 0.0
    last_additional = kpi["t_lastadditionalXPS"]

    print(f"t_firstdirectXPS      (First direct XPS inserted)           :  {first_direct:.2f} min")
    print(f"t_lastdirectXPS       (Last direct XPS inserted)            :  {last_direct:.2f} min")
    print(f"t_firstadditionalXPS  (First additional-check inserted)     :  {first_additional:.2f} min")
    print(f"t_lastadditionalXPS   (Last additional-check inserted)      :  {last_additional:.2f} min\n")

    print("Direct accepted flow times (truck arrival -> inserted):")
    print(f"  Max (t_dir_max)     :  {d_max:.2f} min")
    print(f"  Min (t_dir_min)     :  {d_min:.2f} min")
```

```python
    print(f"  Avg (t_dir_avg)      :  {d_avg:.2f} min")
    print(f"  Median (t_dir_med)   :  {d_median:.2f} min")
    print(f"  Std (t_dir_std)      :  {d_std:.2f} min")
    print(f"  N95 (t_dir_n95)      :  {d_n95:.2f} min\n")

    print("Additional-check flow times (truck arrival -> inserted after reopen):")
    print(f"  Max (t_add_max)      :  {a_max:.2f} min")
    print(f"  Min (t_add_min)      :  {a_min:.2f} min")
    print(f"  Avg (t_add_avg)      :  {a_avg:.2f} min")
    print(f"  Median (t_add_med)   :  {a_median:.2f} min")
    print(f"  Std (t_add_std)      :  {a_std:.2f} min")
    print(f"  N95 (t_add_n95)      :  {a_n95:.2f} min\n")

    print("All inserted packages (direct + additional-check):")
    print(f"  Max (t_XPSmax)       :  {o_max:.2f} min")
    print(f"  Min (t_XPSmin)       :  {o_min:.2f} min")
    print(f"  Avg (t_XPSavg)       :  {o_avg:.2f} min")
    print(f"  Median (t_XPSmed)    :  {o_median:.2f} min")
    print(f"  Std (t_XPSstd)       :  {o_std:.2f} min")
    print(f"  N95 (t_XPSN95)       :  {o_n95:.2f} min\n")





# ============================================================
# Print across-run input statistics (NEW)
# ============================================================
def print_input_statistics_new(truck_stats, package_stats, num_iterations):
    print("\n======================================================================")
    print(f"        INPUT STATISTICS ACROSS NEW PROCESS RUNS ({num_iterations} RUNS)")
    print("======================================================================\n")

    print("TRUCKS PER RUN")
    print(f"Maximum trucks in one run        :  {truck_stats['max']:.0f} (Run {truck_stats['max_run']})")
    print(f"Minimum trucks in one run        :  {truck_stats['min']:.0f} (Run {truck_stats['min_run']})")
    print(f"Median trucks per run            :  {truck_stats['median']:.2f}")
    print(f"Average trucks per run           :  {truck_stats['mean']:.2f}")
    print(f"Std. dev. trucks per run         :  {truck_stats['std']:.2f}\n")

    print("PACKAGES PER RUN")
    print(f"Maximum packages in one run      :  {package_stats['max']:.0f} (Run {package_stats['max_run']})")
    print(f"Minimum packages in one run      :  {package_stats['min']:.0f} (Run {package_stats['min_run']})")
    print(f"Median packages per run          :  {package_stats['median']:.2f}")
    print(f"Average packages per run         :  {package_stats['mean']:.2f}")
    print(f"Std. dev. packages per run       :  {package_stats['std']:.2f}")

# ============================================================
# Print averaged KPI results (NEW) over multiple runs
# ============================================================
def print_averaged_kpis_new(avg_kpi, num_iterations):
    print("\n======================================================================")
    print(f"        AVERAGED NEW PROCESS RESULTS OVER {num_iterations} RUNS")
```

```
    print("====================================================================\n")

    print(f"Average trucks arrived                        :  {avg_kpi['T_arrived']:.2f}")
    print(f"Average trucks processed before T1 reopen     :  {avg_kpi['T_beforeT1reopen']:.2f}")
    print(f"Average trucks processed total                :  {avg_kpi['T_processed']:.2f}")
    print(f"Average trucks waiting at reopen              :  {avg_kpi['T_waitinginqueue_at_reopen']:.2f}")
    print(f"Average trucks docked at reopen               :  {avg_kpi['T_currentlydocked_at_reopen']:.2f}\n")

    print(f"Average last unload time                      :  {avg_kpi['t_lastunload']:.2f} min")
    print(f"Average last auto-scan time                   :  {avg_kpi['t_lastXPSscanned']:.2f} min")
    print(f"Average first direct XPS inserted             :  {avg_kpi['t_firstdirectXPS']:.2f} min")
    print(f"Average last direct XPS inserted              :  {avg_kpi['t_lastdirectXPS']:.2f} min")
    print(f"Average first additional XPS inserted         :  {avg_kpi['t_firstadditionalXPS']:.2f} min")
    print(f"Average last additional XPS inserted          :  {avg_kpi['t_lastadditionalXPS']:.2f} min")
    print(f"Average total process time                    :  {avg_kpi['t_totalprocess']:.2f} min\n")

    print(f"Average truck waiting time                    :  {avg_kpi['avg_truck_waiting_time']:.2f} min")
    print(f"Average truck waiting N95                     :  {avg_kpi['n95_truck_waiting_time']:.2f} min\n")

    print(f"Average XPS spawned total                     :  {avg_kpi['XPS_spawned_total']:.2f}")
    print(f"Average XPS scanned total                     :  {avg_kpi['XPS_scanned_total']:.2f}")
    print(f"Average XPS inserted total                    :  {avg_kpi['XPS_total']:.2f}")
    print(f"Average XPS accepted direct                   :  {avg_kpi['XPS_acceptable']:.2f}")
    print(f"Average XPS additional check                  :  {avg_kpi['XPS_additional_check']:.2f}")
    print(f"Average XPS too early                         :  {avg_kpi['XPS_too_early']:.2f}")
    print(f"Average XPS rejected to T3                    :  {avg_kpi['XPS_rejected_to_T3']:.2f}\n")

    print(f"Average XPS inserted by t = 480               :  {avg_kpi['XPS_inserted_by_reopen']:.2f}")
    print(f"Average XPS inserted after t = 480            :  {avg_kpi['XPS_inserted_after_reopen']:.2f}\n")


    # ------------------------------------------------
    # Direct flow statistics
    # ------------------------------------------------
    print(f"Average mean direct flow time                 :  {avg_kpi['avg_direct_flow_time']:.2f} min")
    print(f"Average median direct flow time               :  {avg_kpi['
```

```
median_direct_flow_time']:.2f} min")
    print(f"Average N95 direct flow time               :  {avg_kpi['n95_direct_flow_time']:.2f} min")
    print(f"Average min direct flow time               :  {avg_kpi['min_direct_flow_time']:.2f} min")
    print(f"Average max direct flow time               :  {avg_kpi['max_direct_flow_time']:.2f} min")
    print(f"Average std. dev. direct flow time         :  {avg_kpi['std_direct_flow_time']:.2f} min\n")

    # ------------------------------------------------
    # Additional-check flow statistics
    # ------------------------------------------------
    print(f"Average mean additional flow time          :  {avg_kpi['avg_additional_flow_time']:.2f} min")
    print(f"Average median additional flow time        :  {avg_kpi['median_additional_flow_time']:.2f} min")
    print(f"Average N95 additional flow time           :  {avg_kpi['n95_additional_flow_time']:.2f} min")
    print(f"Average min additional flow time           :  {avg_kpi['min_additional_flow_time']:.2f} min")
    print(f"Average max additional flow time           :  {avg_kpi['max_additional_flow_time']:.2f} min")
    print(f"Average std. dev. additional flow time     :  {avg_kpi['std_additional_flow_time']:.2f} min\n")

    # ------------------------------------------------
    # Total flow statistics
    # ------------------------------------------------
    print(f"Average mean total flow time               :  {avg_kpi['avg_total_flow_time']:.2f} min")
    print(f"Average median total flow time             :  {avg_kpi['median_total_flow_time']:.2f} min")
    print(f"Average N95 total flow time                :  {avg_kpi['n95_total_flow_time']:.2f} min")
    print(f"Average min total flow time                :  {avg_kpi['min_total_flow_time']:.2f} min")
    print(f"Average max total flow time                :  {avg_kpi['max_total_flow_time']:.2f} min")
    print(f"Average std. dev. total flow time          :  {avg_kpi['std_total_flow_time']:.2f} min\n")
```

C.1.5. process_current.py

```
# ============================================================
# process_current.py
# Refactored CURRENT process (T3 -> T1)
# Ensures consistent KPI counting (T_arrived, T_beforeT1reopen, T_currentlydocked, T_waitinginqueue, T_processed)
# ============================================================

import simpy
import random

from parameters import (
    SIMULATION_END_CURRENT,
    UNLOADING_TIME_CURRENT,
    XPS_ACCEPTED_CURRENT,
    XPS_SHC_CURRENT,
    XPS_REJECTED_CURRENT,
    T3_SCAN_TIME_CURRENT,
```

```python
    T1_SORT_TIME_CURRENT
)

from entities import Truck, XPSPackage

# ============================================================
# STEP 1-4: TRUCK PROCESS
# ============================================================
def truck_process(env, truck, resources, kpi, belly_cart, active_trucks):
    """
    Handles a truck flow through the CURRENT process (T3 -> T1)

    Counter logic:
    - Arrival      T_arrived +1, T_waitinginqueue +1
    - Start unloading     T_waitinginqueue -1, T_currentlydocked +1
    - Finish unloading     T_currentlydocked -1, T_processed +1
    - Trucks that haven't started unloading by SIMULATION_END_CURRENT remain in queue
    """

    # ------------------------------------------------
    # 1. EVENT     Truck arrives at documentation
    # ------------------------------------------------
    truck.arrival_time_CURRENT = env.now
    active_trucks.append(truck)

    # Update KPIs on arrival
    kpi["T_arrived"] += 1
    kpi["T_waitinginqueue"] += 1   # Truck enters waiting queue

    # ------------------------------------------------
    # 2. BUFFER     Truck waits for dock
    # ------------------------------------------------
    with resources["dock_T3_CURRENT"].request() as req:

        # Track peak dock queue length
        current_queue = len(resources["dock_T3_CURRENT"].queue)
        if current_queue > kpi["peak_dock_queue_length"]:
            kpi["peak_dock_queue_length"] = current_queue

        yield req  # Wait until dock becomes available

        # ------------------------------------------------
        # 3. EVENT     Truck moves from queue to dock
        # ------------------------------------------------
        wait_time = env.now - truck.arrival_time_CURRENT
        kpi["truck_waiting_times"].append(wait_time)

        # Check cutoff: do not start unloading after night shift ends
        if env.now >= SIMULATION_END_CURRENT:
            # Truck will not be unloaded, stays in queue
            if truck in active_trucks:
                active_trucks.remove(truck)
            return

        # Truck starts unloading
        kpi["T_waitinginqueue"] -= 1    # Leaves queue
        kpi["T_currentlydocked"] += 1    # Starts unloading in dock

        # ------------------------------------------------
        # 4. ACTIVITY     Full truck unloading
```

```
        # ------------------------------------------------
        yield env.timeout(UNLOADING_TIME_CURRENT)

        # ------------------------------------------------
        # 5. Truck finished unloading
        # ------------------------------------------------

        # --- Take snapshot at night shift end (t_reopen) ---
        if env.now > SIMULATION_END_CURRENT and not kpi.get("t_reopen_recorded", False):
            kpi["T_currentlydocked_at_reopen"] = kpi["T_currentlydocked"]
            kpi["T_waitinginqueue_at_reopen"] = kpi["T_waitinginqueue"]
            kpi["t_reopen"] = SIMULATION_END_CURRENT
            kpi["t_reopen_recorded"] = True

        # --- Now update counters normally ---
        kpi["T_currentlydocked"] -= 1    # Leaves dock
        kpi["T_processed"] += 1           # Count as processed

        # Update night shift KPI if finished before T1 reopens
        if env.now <= SIMULATION_END_CURRENT:
            kpi["T_beforeT1reopen"] += 1

        # Update last unload time
        if env.now > kpi["t_lastunload"]:
            kpi["t_lastunload"] = env.now

        # ------------------------------------------------
        # 6. ACTIVITY & ROUTING     Scan & classify packages
        # ------------------------------------------------
        for i in range(truck.num_packages):
            env.process(
                package_process(
                    env,
                    truck.id,
                    i,
                    truck.arrival_time_CURRENT,   #     pass truck arrival time
                    resources,
                    kpi,
                    belly_cart
                )
            )

    # ------------------------------------------------
    # 7. Remove truck from active list
    # ------------------------------------------------
    if truck in active_trucks:
        active_trucks.remove(truck)


# ==========================================================
# PACKAGE PROCESS     Steps 5  7
# ==========================================================
def package_process(env, truck_id, package_id, package_arrival_time, resources, kpi, belly_cart):
    arrival_time = package_arrival_time

    # ------------------------------------------------
    # 5. ACTIVITY     Human verification
    # ------------------------------------------------
    with resources["T3_handler_CURRENT"].request() as req:
```

```
            yield req
            yield env.timeout(T3_SCAN_TIME_CURRENT)

        # Record last scanned package time
        kpi["t_lastXPSscanned"] = max(
            kpi.get("t_lastXPSscanned", 0),
            env.now
        )

        # Track total packages scanned
        kpi["XPS_scanned_total"] = kpi.get("XPS_scanned_total", 0) + 1

        # ------------------------------------------------
        # 6. ROUTING     Classify package
        # ------------------------------------------------
        rnd = random.random()
        if rnd < XPS_ACCEPTED_CURRENT:
            pkg_type = "accepted"
            kpi["XPS_acceptable"] += 1
            kpi["XPS_total"] += 1           # Only accepted packages enter sorter
            yield belly_cart.put({"arrival_time": arrival_time, "type": pkg_type})
        elif rnd < XPS_ACCEPTED_CURRENT + XPS_SHC_CURRENT:
            pkg_type = "SHC"
            kpi["XPS_SHC"] += 1
        else:
            pkg_type = "rejected"
            kpi["XPS_unacceptable"] += 1

        # Placeholder for flow time; updated when scanned into sorter
        # kpi["XPS_flow_times"].append(0)


# ============================================================
# SORTER WORKER     Step 8
# ============================================================
def sorter_worker_current(env, worker_id, kpi, belly_cart):
    """
    One T1 sorter worker processes accepted packages from the belly cart.
    Multiple workers can run in parallel.
    """
    while True:
        if len(belly_cart.items) == 0:
            if kpi.get("sorting_ready", False):
                break
            yield env.timeout(1)
            continue

        pkg = yield belly_cart.get()
        yield env.timeout(T1_SORT_TIME_CURRENT)

        if kpi["t_firstXPSinserted"] is None:
            kpi["t_firstXPSinserted"] = env.now

        flow_time = env.now - pkg["arrival_time"]
        kpi["XPS_flow_times"].append(flow_time)

        kpi["t_lastXPSinserted"] = env.now
        kpi["t_totalprocess"] = env.now


# ============================================================
```

```
# SORTER PROCESS     Step 8
# ============================================================
def sorter_process(env, resources, kpi, belly_cart, active_trucks):
    """
    In the current process, accepted XPS packages wait in the belly-cart
    buffer until T1 reopens. After T1 reopens, any trucks still being
    processed may finish unloading and their packages may still finish
    scanning. Only then can accepted packages be inserted into the sorter.
    """

    # Wait until T1 reopens
    if env.now < SIMULATION_END_CURRENT:
        yield env.timeout(SIMULATION_END_CURRENT - env.now)

    # After reopen, wait until all active trucks are done and all T3 scanning is finished
    while (
        len(active_trucks) > 0
        or resources["T3_handler_CURRENT"].count > 0
        or len(resources["T3_handler_CURRENT"].queue) > 0
    ):
        yield env.timeout(1)

    # Signal that no more accepted packages will be added
    kpi["sorting_ready"] = True

    # Start one sorter worker per available T1 handler
    for worker_id in range(resources["T1_handler_CURRENT"].capacity):
        env.process(sorter_worker_current(env, worker_id, kpi, belly_cart))
```

C.1.6. run_simulation_current.py

```
# ============================================================
# run_simulation_current.py
# Refactored runner for the CURRENT process
# ============================================================

import os
import simpy
import numpy as np

from entities import Truck
from resources import create_resources
from process_current import truck_process, sorter_process
from parameters import SIMULATION_END_CURRENT, NUM_ITERATIONS
from kpi_tracking import (
    initialize_kpis,
    print_kpis_current,
    summarize_kpis_current,
    average_kpi_summaries_current,
    print_averaged_kpis_current,
    summarize_input_statistics,
    print_input_statistics_current,
)
from arrival_data import load_interarrival_times, mean_interarrival, load_package_counts


# ============================================================
# TRUCK GENERATOR
# ============================================================
```

```python

def truck_generator(env, rng_generator, resources, kpi, belly_cart, active_trucks,
                    mean_interarrival_time, package_counts):
    """
    Generates trucks arriving until the simulation end time.

    Truck interarrival times are sampled from an exponential distribution
    using the observed mean interarrival time.

    Packages per truck are sampled from the observed empirical distribution.
    """
    truck_id = 0

    while True:
        # Sample time until next truck
        next_truck_time = int(round(rng_generator.exponential(
            mean_interarrival_time)))

        # Prevent zero-minute interarrival times
        if next_truck_time < 1:
            next_truck_time = 1

        # Stop if next truck would arrive after the night window
        if env.now + next_truck_time >= SIMULATION_END_CURRENT:
            break

        kpi["inter_arrival_times"].append(next_truck_time)
        yield env.timeout(next_truck_time)

        truck_id += 1

        # Sample number of packages from observed data
        packages_inside_truck = int(rng_generator.choice(package_counts))

        kpi["truck_arrival_times"].append(env.now)
        kpi["truck_package_counts"].append(packages_inside_truck)

        truck = Truck(truck_id, packages_inside_truck)

        env.process(
            truck_process(env, truck, resources, kpi, belly_cart, active_trucks)
        )


# ============================================================
# RECORD KPIs AT NIGHT SHIFT END (t_reopen)
# ============================================================

def record_t_reopen(env, kpi):
    """
    Records KPIs exactly at night shift end (t_reopen).
    """
    yield env.timeout(SIMULATION_END_CURRENT)
    kpi["t_reopen"] = SIMULATION_END_CURRENT
    kpi["T_currentlydocked_at_reopen"] = kpi["T_currentlydocked"]
    kpi["T_waitinginqueue_at_reopen"] = kpi["T_waitinginqueue"]


# ============================================================
# MAIN SIMULATION FUNCTION
# ============================================================
```

```
def run_simulation_current(print_results=True):
    """
    Sets up the simulation environment, resources, KPIs, and starts
    the truck generator and sorter process for CURRENT process.
    """
    rng_generator = np.random.default_rng()

    # Get folder where this script is located
    base_dir = os.path.dirname(os.path.abspath(__file__))

    # Build full paths to input files
    interarrival_file = os.path.join(base_dir, "interarrival_times.csv")
    packages_file = os.path.join(base_dir, "packages_per_truck.csv")

    # Load observed data
    interarrival_times = load_interarrival_times(interarrival_file)
    mean_interarrival_time = mean_interarrival(interarrival_times)

    package_counts = load_package_counts(packages_file)

    env = simpy.Environment()

    # -----------------------------
    # Resources
    # -----------------------------
    resources = create_resources(env)

    # -----------------------------
    # KPI tracking
    # -----------------------------
    kpi = initialize_kpis()

    # -----------------------------
    # Shared structures
    # -----------------------------
    belly_cart = simpy.Store(env)
    active_trucks = []

    # -----------------------------
    # Start truck arrivals
    # -----------------------------
    env.process(
        truck_generator(
            env,
            rng_generator,
            resources,
            kpi,
            belly_cart,
            active_trucks,
            mean_interarrival_time,
            package_counts
        )
    )

    # -----------------------------
    # Record t_reopen snapshot
    # -----------------------------
    env.process(record_t_reopen(env, kpi))

    # -----------------------------
```

```
    # Start sorter process
    # ----------------------------
    env.process(sorter_process(env, resources, kpi, belly_cart, active_trucks))

    # ----------------------------
    # Run simulation
    # ----------------------------
    env.run()

    # ----------------------------
    # Print KPIs for one run
    # ----------------------------
    if print_results:
        print_kpis_current(kpi)

    return kpi


# ============================================================
# MAIN
# ============================================================

if __name__ == "__main__":

    N_RUNS = NUM_ITERATIONS
    all_kpi_summaries = []

    truck_counts = []
    package_counts = []

    for i in range(N_RUNS):
        print("\n====================================================")
        print(f"RUN {i+1} / {N_RUNS}")
        print("====================================================")

        kpi = run_simulation_current(print_results=True)

        # KPI summary for averaged performance results
        kpi_summary = summarize_kpis_current(kpi)
        all_kpi_summaries.append(kpi_summary)

        # Across-run input statistics
        truck_counts.append(kpi["T_arrived"])
        package_counts.append(kpi["XPS_scanned_total"])

    # Across-run input statistics
    truck_stats = summarize_input_statistics(truck_counts)
    package_stats = summarize_input_statistics(package_counts)
    print_input_statistics_current(truck_stats, package_stats, N_RUNS)

    # Averaged KPI results
    averaged_kpis = average_kpi_summaries_current(all_kpi_summaries)
    print_averaged_kpis_current(averaged_kpis, N_RUNS)
```

C.1.7. process_new.py

```
# ============================================================
# process_new.py
# NEW process:
# - Truck uses ONE drop-off machine
# - Unload (UNLOADING_TIME_NEW)
```

```
# - Sequential auto-scan per package at the same machine (AUTO_SCAN_TIME_NEW)
# - Routing:
#     * accepted direct -> immediately "inserted into sorter" (complete)
#     * additional check -> buffered until T1 reopens, then manually inserted by
    T1 handlers
#     * too early / rejected -> counted and exit system (for now)
# ============================================================

import random

from parameters import (
    SIMULATION_END_NEW,
    UNLOADING_TIME_NEW,
    AUTO_SCAN_TIME_NEW,
    NEW_ACCEPTED_DIRECT_NEW,
    NEW_ADDITIONAL_CHECK_NEW,
    NEW_TOO_EARLY_NEW,
    T1_SORT_TIME_NEW,
)


# ============================================================
# TRUCK PROCESS (NEW)
# ============================================================
def truck_process_new(env, truck, resources, kpi, additionalcheck_buffer,
    active_trucks):
    """
    Truck arrives -> waits for drop-off machine -> unloads -> sequential scan+
        route -> depart.
    Cutoff rule: if unloading would START after SIMULATION_END_NEW, truck is
        ignored (no packages).
    """
    truck.arrival_time_NEW = env.now
    active_trucks.append(truck)

    # Arrival KPIs
    kpi["T_arrived"] += 1
    kpi["T_waitinginqueue"] += 1

    with resources["drop_machines_NEW"].request() as req:
        # Track peak queue length at drop-off machines
        current_queue = len(resources["drop_machines_NEW"].queue)
        kpi["peak_drop_queue_length"] = max(kpi["peak_drop_queue_length"],
            current_queue)

        yield req  # machine acquired

        # Waiting time
        wait_time = env.now - truck.arrival_time_NEW
        kpi["truck_waiting_times"].append(wait_time)

        # Cutoff: do not start unloading after night window ends
        if env.now >= SIMULATION_END_NEW:
            kpi["T_waitinginqueue"] -= 1
            if truck in active_trucks:
                active_trucks.remove(truck)
            return

        # Starts unloading
        kpi["T_waitinginqueue"] -= 1
        kpi["T_currentlydocked"] += 1
```

```python

        # Unload
        yield env.timeout(UNLOADING_TIME_NEW)
        kpi["t_lastunload"] = max(kpi["t_lastunload"], env.now)

        # Sequential scan + routing at the same machine
        kpi["XPS_spawned_total"] += truck.num_packages

        for _ in range(truck.num_packages):
            yield env.timeout(AUTO_SCAN_TIME_NEW)

            # Scanned counters + time trigger
            kpi["XPS_scanned_total"] += 1
            kpi["t_lastXPSscanned"] = max(kpi["t_lastXPSscanned"], env.now)

            rnd = random.random()

            if rnd < NEW_ACCEPTED_DIRECT_NEW:
                # Direct accepted -> inserted immediately (complete)
                kpi["XPS_acceptable"] += 1
                kpi["XPS_total"] += 1

                if env.now <= SIMULATION_END_NEW:
                    kpi["XPS_inserted_by_reopen"] += 1
                else:
                    kpi["XPS_inserted_after_reopen"] += 1


                flow_time = env.now - truck.arrival_time_NEW
                kpi["XPS_flow_times_direct"].append(flow_time)
                kpi["XPS_flow_times"].append(flow_time)

                if kpi["t_firstdirectXPS"] is None:
                    kpi["t_firstdirectXPS"] = env.now

                kpi["t_lastdirectXPS"] = max(kpi["t_lastdirectXPS"], env.now)
                kpi["t_totalprocess"] = max(kpi["t_totalprocess"], env.now)

            elif rnd < NEW_ACCEPTED_DIRECT_NEW + NEW_ADDITIONAL_CHECK_NEW:
                # Additional check -> buffer until reopen
                kpi["XPS_additional_check"] += 1
                yield additionalcheck_buffer.put({"arrival_time": truck.
                    arrival_time_NEW})

            elif rnd < NEW_ACCEPTED_DIRECT_NEW + NEW_ADDITIONAL_CHECK_NEW +
                NEW_TOO_EARLY_NEW:
                kpi["XPS_too_early"] += 1

            else:
                kpi["XPS_rejected_to_T3"] += 1
                kpi["XPS_unacceptable"] += 1

        # Truck done, releases machine
        kpi["T_currentlydocked"] -= 1
        kpi["T_processed"] += 1

        if env.now <= SIMULATION_END_NEW:
            kpi["T_beforeT1reopen"] += 1

    if truck in active_trucks:
        active_trucks.remove(truck)
```

```python
# ============================================================
# ADDITIONAL CHECK WORKER (manual insertion after reopen)
# ============================================================
def additional_check_worker_new(env, worker_id, kpi, additionalcheck_buffer, active_trucks):
    """
    One T1 worker processes buffered additional-check packages after reopen.
    """
    # Wait until T1 reopens
    yield env.timeout(SIMULATION_END_NEW)

    while True:
        all_trucks_done = len(active_trucks) == 0
        all_scanned = kpi.get("XPS_scanned_total", 0) >= kpi.get("XPS_spawned_total", 0)
        buffer_empty = len(additionalcheck_buffer.items) == 0
        arrivals_done = kpi.get("arrivals_done", False)

        # Only stop when the whole upstream process is truly finished
        if arrivals_done and all_trucks_done and all_scanned and buffer_empty:
            break

        # If buffer is empty, wait and check again
        if buffer_empty:
            yield env.timeout(1)
            continue

        # Take exactly one package
        pkg = yield additionalcheck_buffer.get()

        # Process exactly one package
        yield env.timeout(T1_SORT_TIME_NEW)

        # Package is now inserted into sorter
        kpi["XPS_total"] += 1

        if env.now <= SIMULATION_END_NEW:
            kpi["XPS_inserted_by_reopen"] += 1
        else:
            kpi["XPS_inserted_after_reopen"] += 1


        flow_time = env.now - pkg["arrival_time"]
        kpi["XPS_flow_times_additional"].append(flow_time)
        kpi["XPS_flow_times"].append(flow_time)

        if kpi["t_firstadditionalXPS"] is None:
            kpi["t_firstadditionalXPS"] = env.now

        kpi["t_lastadditionalXPS"] = max(kpi["t_lastadditionalXPS"], env.now)
        kpi["t_totalprocess"] = max(kpi["t_totalprocess"], env.now)



def process_additional_check_package(env, pkg, resources, kpi):
    """
    Process one buffered additional-check package after reopen.
    Multiple instances of this process can run in parallel.
```

```
    """
    with resources["T1_handler_NEW"].request() as req:
        yield req
        yield env.timeout(T1_SORT_TIME_NEW)

    # Now considered inserted/complete
    kpi["XPS_total"] += 1

    flow_time = env.now - pkg["arrival_time"]

    kpi["XPS_flow_times_additional"].append(flow_time)
    kpi["XPS_flow_times"].append(flow_time)

    # Record first additional-check insertion
    if kpi["t_firstadditionalXPS"] is None:
        kpi["t_firstadditionalXPS"] = env.now

    kpi["t_lastadditionalXPS"] = max(kpi["t_lastadditionalXPS"], env.now)
    kpi["t_totalprocess"] = max(kpi["t_totalprocess"], env.now)



# ============================================================
# END CONDITION MONITOR (NEW)
# ============================================================
def end_condition_monitor_new(env, kpi, active_trucks, additionalcheck_buffer,
    resources, done_event):
    """
    Ends simulation once:
    - arrivals are done
    - no trucks remain active
    - all spawned packages have completed auto-scan
    - additionalcheck buffer is empty
    - T1 handlers are idle
    """
    while True:
        if (
            kpi.get("arrivals_done", False)
            and len(active_trucks) == 0
            and kpi.get("XPS_scanned_total", 0) >= kpi.get("XPS_spawned_total", 0)
            and len(additionalcheck_buffer.items) == 0
            and resources["T1_handler_NEW"].count == 0
        ):
            if not done_event.triggered:
                done_event.succeed()
            return

        yield env.timeout(1)
```

C.1.8. run_simulation_new.py

```
# ============================================================
# run_simulation_new.py
# Runner for the NEW process:
# - direct accepted completes immediately after auto-scan
# - additional-check packages buffered until reopen, then manually inserted
# ============================================================

import os
import simpy
import numpy as np
```

```python

from entities import Truck
from resources import create_resources

from parameters import SIMULATION_END_NEW, NUM_ITERATIONS, T1_HANDLER_COUNT_NEW

from process_new import (
    truck_process_new,
    additional_check_worker_new,
    end_condition_monitor_new
)

from kpi_tracking import (
    initialize_kpis_new,
    print_kpis_new,
    summarize_kpis_new,
    average_kpi_summaries_new,
    print_averaged_kpis_new,
    summarize_input_statistics,
    print_input_statistics_new,
)
from arrival_data import load_interarrival_times, mean_interarrival,
    load_package_counts


# ============================================================
# TRUCK GENERATOR
# ============================================================
def truck_generator_new(env, rng_generator, resources, kpi, additionalcheck_buffer
    , active_trucks,
                        mean_interarrival_time, package_counts):
    truck_id = 0

    while True:
        # Sample time until next truck and make it an integer
        next_truck_time = int(round(rng_generator.exponential(
            mean_interarrival_time)))

        # Prevent zero-minute interarrival times
        if next_truck_time < 1:
            next_truck_time = 1

        # Stop if next truck would arrive after the night window
        if env.now + next_truck_time >= SIMULATION_END_NEW:
            break

        # Store interarrival time and wait for next truck
        kpi["truck_interarrival_times"].append(next_truck_time)
        yield env.timeout(next_truck_time)

        truck_id += 1

        # Sample number of packages from observed data
        packages_in_truck = int(rng_generator.choice(package_counts))

        # Store debug info
        kpi["truck_arrival_times"].append(env.now)
        kpi["truck_package_counts"].append(packages_in_truck)

        truck = Truck(truck_id, packages_in_truck)

```

```
        env.process(
            truck_process_new(env, truck, resources, kpi, additionalcheck_buffer,
                active_trucks)
        )

    # Tell monitor no more trucks will arrive
    kpi["arrivals_done"] = True


# ============================================================
# RECORD SNAPSHOT AT t_reopen
# ============================================================
def record_t_reopen_new(env, kpi):
    yield env.timeout(SIMULATION_END_NEW)
    kpi["t_reopen"] = SIMULATION_END_NEW
    kpi["T_currentlydocked_at_reopen"] = kpi["T_currentlydocked"]
    kpi["T_waitinginqueue_at_reopen"] = kpi["T_waitinginqueue"]


# ============================================================
# MAIN SIMULATION
# ============================================================
def run_simulation_new(print_results=True):
    # Random number generator (no seed)
    rng_generator = np.random.default_rng()

    # Load observed data
    base_dir = os.path.dirname(os.path.abspath(__file__))

    interarrival_file = os.path.join(base_dir, "interarrival_times.csv")
    packages_file = os.path.join(base_dir, "packages_per_truck.csv")

    interarrival_times = load_interarrival_times(interarrival_file)
    mean_interarrival_time = mean_interarrival(interarrival_times)

    package_counts = load_package_counts(packages_file)

    env = simpy.Environment()

    # Resources
    resources = create_resources(env)

    # KPI tracking
    kpi = initialize_kpis_new()

    # Buffers
    additionalcheck_buffer = simpy.Store(env)

    # Active truck list (for termination logic)
    active_trucks = []

    # Termination event
    done_event = env.event()

    # Start processes
    env.process(
        truck_generator_new(
            env,
            rng_generator,
            resources,
            kpi,
```

```
                additionalcheck_buffer,
                active_trucks,
                mean_interarrival_time,
                package_counts
            )
        )

    env.process(record_t_reopen_new(env, kpi))

    # Manual insertion process after reopen
    for worker_id in range(T1_HANDLER_COUNT_NEW):
        env.process(
            additional_check_worker_new(
                env,
                worker_id,
                kpi,
                additionalcheck_buffer,
                active_trucks
            )
        )

    # Termination monitor
    env.process(
        end_condition_monitor_new(
            env,
            kpi,
            active_trucks,
            additionalcheck_buffer,
            resources,
            done_event
        )
    )

    # Run simulation until monitor triggers completion
    env.run(until=done_event)

    # Print results for one run if wanted
    if print_results:
        print_kpis_new(kpi)

    return kpi


# ============================================================
# MAIN
# ============================================================
if __name__ == "__main__":

    N_RUNS = NUM_ITERATIONS
    all_kpi_summaries = []

    truck_counts = []
    package_totals_per_run = []

    for i in range(N_RUNS):
        print("\n==================================================")
        print(f"RUN {i+1} / {N_RUNS}")
        print("==================================================")

        kpi = run_simulation_new(print_results=True)

```

```
        # KPI summary for averaged performance results
        kpi_summary = summarize_kpis_new(kpi)
        all_kpi_summaries.append(kpi_summary)

        # Across-run input statistics
        truck_counts.append(kpi["T_arrived"])
        package_totals_per_run.append(kpi["XPS_spawned_total"])

    # Across-run input statistics
    truck_stats = summarize_input_statistics(truck_counts)
    package_stats = summarize_input_statistics(package_totals_per_run)
    print_input_statistics_new(truck_stats, package_stats, N_RUNS)

    # Averaged KPI results
    averaged_kpis = average_kpi_summaries_new(all_kpi_summaries)
    print_averaged_kpis_new(averaged_kpis, N_RUNS)
```

C.1.9. arrival_data.py

```
# ============================================================
# arrival_data.py
# ============================================================

import csv


def load_interarrival_times(csv_path):
    """Reads a CSV with one column of interarrival times (minutes)."""
    times = []
    with open(csv_path, "r", newline="") as f:
        reader = csv.reader(f)
        next(reader, None)  # skip header row

        for row in reader:
            if not row:
                continue
            try:
                t = float(row[0])
                if t > 0:
                    times.append(t)
            except ValueError:
                continue

    return times


def mean_interarrival(times):
    """Returns the average interarrival time."""
    if len(times) == 0:
        raise ValueError("No valid interarrival times found.")
    return sum(times) / len(times)


def load_package_counts(csv_path):
    """Reads a CSV with one column of package counts per truck."""
    counts = []
    with open(csv_path, "r", newline="") as f:
        reader = csv.reader(f)
        next(reader, None)  # skip header row

        for row in reader:
```

```
            if not row:
                continue
            try:
                value = int(float(row[0]))
                if value >= 0:
                    counts.append(value)
            except ValueError:
                continue

    return counts
```

D

Appendix D

D.1. Anonymous summaries of interviews

For information gathering and insight into the KLM Cargo processes, nine individuals were interviewed (see section 3.2.4). For each of these interviews, a condensed, anonymous summary was created representing the core information obtained. These summaries are listed below from D.1.1 to D.1.9:

D.1.1. Interview 1 of 9

Interviewee Experience

- Managerial role in Terminals 2 and 3, focused on import and export processes of general cargo
- 1.5 years in current role
- Over 25 years of experience in the air cargo industry
- Indirect involvement in operations through coordination, planning, communication, and oversight

Problems / Bottlenecks

- XPS packages must be handled using the same equipment (forklifts) as general cargo, despite being smaller and lighter
- Nighttime is the busiest operational period
- Trucks assigned to T3 experience waiting times of several hours
- Express drivers must queue at night and require additional time to offload
- Driver dissatisfaction due to delays
- Lack of strict agreements or standardized guidelines for these processes
- No clear communication between T1 and T2/T3
  - Inconsistent information regarding operations and procedures
- One of the primary bottlenecks is excessive waiting times at the loading doors

Desirables / Goals

- Vans and trucks carrying only a few packages should no longer need to go to T3
- Retain the positive aspect of 24/7 operations
- Enable express package drop-off at T1 to reduce congestion and workload at T3
- Quick and streamlined processing
- High reliability
- Compliance with customs regulations
- Ability to identify and mitigate safety and security risks

Stakeholders

- Truck companies and freight agents
- Drivers waiting in queues
- KLM Cargo

Drop-off Solution

- Overall positive view of the proposed solution
- Considered a major operational improvement if implemented
- Strong alignment with the express product offering
- Potentially highly effective for both drivers and KLM Cargo
- Concerns include safety and security risks (e.g., damaged packages)
- Physical inspection procedures may pose implementation challenges

Additional Notes

- Build-up and breakdown operations handle transit flow and are not managed by Impex
- Impex focuses on truck loading and offloading processes
- Cargo is accepted once documentation is complete and ready for carriage

D.1.2. Interview 2 of 9

Interviewee Experience

- Project Manager at KLM Cargo, focused on EPS processes
- 3 years of experience in current role
- Involved in the transition from Chain to iCargo
- Indirect involvement in operations through coordination, planning, communication, and oversight

Problems / Bottlenecks

- Operational issues occur within "Dwars Transport" (movement of shipments between T1 and T2/T3)
- Shipments are occasionally misplaced or stored in incorrect locations
- In iCargo, shipments in transit are broadly labeled as "Dwars Transport," limiting visibility into their exact status
- During peak nights with high XPS volumes, up to eight carts of shipments may accumulate at T1 by morning, creating congestion at the Oogplein
- Morning staff retrieving cargo do not always check iCargo in advance to verify expected shipments, increasing the risk of unnoticed missing packages
- Limited prioritization of shipments; XPS cargo is handled similarly to general cargo despite higher urgency
- Night-time acceptance of human remains shipments for European destinations was discontinued due to repeated process failures

Desirables / Goals

- Clear differentiation between cargo types (e.g., Dangerous Goods and other special handling codes)
- Reduce pressure on the Dwars Transport process and increase its robustness
- Enable prioritization of shipments based on flight deadlines
  - Shipments for earlier departures processed first

- Shipments with longer lead times processed later
- User-friendly system design, particularly for truck drivers who prefer direct human interaction
- Ability to address potential language barriers
- Availability of a backup system ("Plan B")
- Clear overview of exceptions that cannot be accepted at the drop-off point

Stakeholders

- T1 employees
- Unit Manager for T1

Drop-off Solution

- Viewed as having strong potential to reduce current process errors
- Must adequately address operational concerns and exception handling to be successful

Additional Notes

- The term "deliveries" can be ambiguous, as it is used differently for import and export processes
- In the iCargo system:
  - "Delivery" refers to import-related handling
  - "Acceptance" refers to export-related handling (e.g., night-time acceptance of shipments)
- Transfer of cargo from T3 to T1 is performed by a T1 employee
- The Transport department moves shipments from the platform to the AL-banen
- XPS has a maximum weight limit of 23 kg; shipments exceeding this are designated XPH ("Express Heavy")
- If the weight listed on the FWB matches the actual weight, the shipment is processed at T3 and built onto a ULD
- M21 is the code for XPS; M25 is the code for XPH
- "Happy flows" refer to standard processes without exceptions, whereas "unhappy flows" require special handling or exception management

D.1.3. Interview 3 of 9

Interviewee Experience

- Managerial role related to ground handling processes at the terminals
- Acts as a bridge between customers/clients and operational departments
- Over 30 years of experience at KLM
- 7 years in current role
- Indirect involvement in operations through coordination, planning, communication, and oversight

Problems / Bottlenecks

- The current process is identified as a root cause of poor flown-as-planned performance
- Terminal 1 is closed during nighttime operations
- The only advantage of the current process is that packages can technically still be delivered 24/7
- The existing setup was originally introduced as a temporary solution after Terminal 1 stopped operating 24/7
- Vans and trucks delivering XPS shipments at night (even with only 1–2 packages) must use the same facilities as general cargo deliveries, increasing traffic, congestion, and waiting times

- Packages delivered at night do not enter the sorter until Terminal 1 reopens (between 06:00–08:00), resulting in lost processing time
- Early morning flight departures are missed due to delays
- Miscommunication between T1 and T2/T3 personnel causes further delays in the transfer process after reopening
  - Transfers may occur at 07:00 or 07:30 instead of immediately at 06:00

Desirables / Goals

- Ensure timely transfer of shipments to T1 with clear control mechanisms
- Implement system alerts if shipments have not reached T1 before a defined deadline (e.g., 15 minutes prior to cutoff)
- Integrate monitoring tools for shift leaders or team coordinators
- Eliminate lateral transport of shipments through an improved design
- Weatherproof solution
- Preferably integrated into the exterior wall of T1 with direct connection to the sorter
- Capability to process IATA-standardized labels
- Clear procedures for handling XPH, oversized shipments, and cargo requiring special handling
- Ability to separate shipments requiring additional screening or security checks
- Clear refusal-handling procedure to prevent customers from leaving rejected shipments on-site

Stakeholders

- Customers / Shippers (e.g., DSV, HML)
- Clients (e.g., ASML)
- KLM Cargo
- KLM Cargo Operations
- KLM Cargo BPI (Business Process Improvement)
- KLM Cargo S&D (Sales and Distribution – commercial team)
- Product engineers and product owners for the express product
- Compliance department

Drop-off Solution

- Stakeholders such as drivers, service providers, and 3PLs are familiar with similar automated drop-off systems
- Considered user-friendly and easy to implement, particularly given the small and lightweight nature of XPS shipments
- Optimal placement would be integrated into the wall of T1 near the export acceptance area

Additional Notes

- Interviewee originated the drop-off solution concept
- Terminal 1 previously operated 24/7 but nighttime closure was implemented to reduce operational costs
- Potential use of RFID technology to improve shipment tracking, process control, and transparency
- Upon offloading, cargo is labeled FOH (Freight on Hand), after which acceptance checks are performed (e.g., damage inspection)

D.1.4. Interview 4 of 9

Interviewee Experience

- Managerial role related to EPS processes and postal operations
- Subject matter expert in EPS and Terminal 1 procedures and operations
- 13 years of experience at KLM
- 4 years in current role
- Indirect involvement in operations through coordination, planning, communication, and oversight

Problems / Bottlenecks

- Shipments are not organized by priority (e.g., cargo for a 09:00 flight and a 16:00 flight may be placed on the same cart and processed randomly upon arrival at T1)
- T1 Oogplein frequently overloaded during peak periods, with 6–10 full belly carts of unsorted EPS packages occupying significant physical space
- Inconsistency regarding responsibility for transporting belly carts from T3 to T1
  - Sunday to Monday: performed by T3 employees
  - Saturday: performed by T1 employees
- Limited tracking visibility during Dwars Transport; shipments share a single location status until scanned at T1
- XPS shipments occasionally accepted at T3 despite not meeting criteria (e.g., exceeding 23 kg, containing dry ice)
- Late handling results in shipments missing early morning departures
- Multiple handover moments increase the risk of process failure or lost shipments
- Occurrence of lost packages

Desirables / Goals

- Reduce spatial congestion at the Oogplein
- Minimize the need for night shifts where possible
- Enable autonomous acceptance of parcels during nighttime operations (and potentially daytime)
- Prefer a solution directly connected to the sorter, avoiding intermediate storage
- Prefer system compatibility with the current sorter developer for seamless integration
- Include an emergency stop function
- Avoid increasing physical labor demands (in compliance with Dutch labor inspection regulations)
- Establish a clear backup procedure in case of system failure
- Prevent drivers from bypassing T1 and delivering directly to T3

Stakeholders

- EPS Unit (Express and Postal Solutions – receiving party)
- Customers (delivering party)
  - Transport companies / freight forwarders
- Sales and Distribution Department
- Compliance department
- Product Owner Team (Industrial Means)
- iCargo team

Drop-off Solution

- A stricter autonomous solution may increase revenue through improved weight-limit enforcement

- Potential to reduce staffing needs and workload if implemented during daytime operations as well
- Strongly positive view of the proposed solution
- Expected to reduce time, labor effort, and spatial congestion in Terminal 3
- Key challenge will be compliance with acceptance criteria as outlined in the Cargo Handling Manual
- Preferred location near the sorter connection to the T1 Voorloods

Additional Notes

- Previous latest acceptance time was 90 minutes before flight departure
- High operational cost of keeping T1 open at night for limited shipment volume
- Night shifts are expensive and negatively impact employee health, contributing to fatigue-related errors and increased sickness levels
- Express and mail products are subject to selective loading procedures
- XPH, fresh, and pharma shipments arriving at T1:
  - European-dedicated shipments handled at T1
  - Intercontinental shipments transferred to T2
- Import shipments:
  - Transferred to T2 via Dwars Transport
  - Not processed during nighttime operations
  - Potential future research opportunity: assess feasibility of nighttime pickup through the proposed solution
- Shipments intended for early morning departures (08:30 / 09:30) must be delivered by 21:00 the previous evening

D.1.5. Interview 5 of 9

Interviewee Experience

- Directorial and oversight role focused on monitoring and improving KLM Cargo business processes
- 15 years of experience at KLM
- 3 years in current role
- Indirect involvement in operations through coordination, planning, communication, and oversight

Problems / Bottlenecks

- The previous system (Chain) provided more detailed tracking information (e.g., chute location and time window), which has been reduced in iCargo to a single status location during the T3–T1 process
- T2 Dwarsbuffer presents some risk of lost parcels, though relatively limited
- T3 XPS temporary storage has experienced more frequent issues with lost parcels
- Transporting trains of belly carts from T3 through the warehouse to T1 presents safety concerns
- Inefficient use of space in the T3 Voorloods, where small quantities of XPS packages occupy wooden pallets in an already constrained area
- No automated trigger for warehouse employees to collect XPS shipments in T3, resulting in variable pickup times (e.g., 05:45 or 07:00)
- Oogplein becomes overloaded during peak periods
- Misalignment in acceptance processes: T1 is dedicated to XPS acceptance, while T3 accepts all cargo types, resulting in inconsistent customer experience

- Export acceptance quality at T3 decreases when shipments require transit to T1
- The sorter system is currently not utilized during nighttime operations

Desirables / Goals

- Maintain the ability for XPS packages to be delivered during nighttime
- Fully automated process without requiring night staff
- Automatic scanning, weighing, and validation of parcels against the FWB prior to acceptance
- Potential implementation during daytime operations as well
- Ability to handle varying night-time delivery volumes
- Immediate support availability from the documentation department in case of system issues
- Direct physical and digital integration with the sorter system
- Implementation of piece-level 3D scanning
- Legal-for-trade certified weighing and volumetric measurement
- Allow human staff to focus on non-standard shipments during daytime if automation is extended
- Automated acceptance checks
  - Potential investment in X-ray scanning technology

Stakeholders

- Express forwarders (primary express customers)
  - Subset currently utilizing night delivery services
  - KLM Cargo works directly with forwarders
- Couriers
- Internal stakeholders
  - EPS department
  - Hub Manager
  - Cargo Control Center management
  - Operational Integrity, Compliance, and Safety department
- Operational support and external suppliers (e.g., Lödige)
- KLM Cargo Digital department representative
- Shippers (indirectly affected)

Drop-off Solution

- Strongly positive attitude toward the proposed solution
- High confidence in the concept
- Acceptance criteria (weight, dimensions, LAT) are clear and transparent
- Main challenge lies in integration within the KLM Cargo digital landscape
- Significant testing would be required prior to implementation
- Expected to improve customer and driver satisfaction by reducing waiting times

Additional Notes

- ULDs from A321neo aircraft are offloaded at the AL-banen
- Dedicated buffer area in T2 referred to as "DWARS"
- APH containers originate from A321 aircraft
- AKE (LD3) containers are lower-deck units used on widebody aircraft
- Shipment flow:

  - Shipper delivers package to express forwarder
  - Express forwarder (e.g., DHL, UPS) transports shipments to KLM Cargo using vans or small lorries

D.1.6. Interview 6 of 9

Interviewee Experience

- Employee involved in operations, procedures, and commercial aspects of express products
- Over 10 years of experience at KLM
- Indirect involvement in operations through coordination, planning, communication, and oversight

Problems / Bottlenecks

- Storage of XPS packages on belly carts occupies space in T2/T3, interfering with regular warehouse operations
- Delivering XPS shipments to T3 introduces additional procedural steps and protocols, negatively impacting performance
- Delays in collecting shipments from T3 result in missed connections and booked flights
- Shipment prioritization is not consistently applied
- Process steps are not always executed in a timely manner
- Frequent deviations between actual shipments and original bookings

Desirables / Goals

- Enable direct delivery of XPS shipments to T1
- Improve prioritization of shipments
- Ensure shipments depart on their intended flights
- Guarantee accurate measurement, weighing, and compliance with acceptance criteria
- Establish procedures to manage shipments that deviate from original bookings

Stakeholders

- Clients
- Customers
- EPS department
- Warehouse staff / operational personnel

Drop-off Solution

- Strong support for implementation of such a solution
- System must include integrated weighing scale and volumetric measurement capability

Additional Notes

- Storage location for XPS in T2 is referred to as "DWXPS"
- Package dimension limits are determined by aircraft door constraints
- Maximum acceptable dimensions: $130 \times 80 \times 80$ cm
- Shipments exceeding 23 kg or dimension limits must be processed in T3

D.1.7. Interview 7 of 9

Interviewee Experience

- Coordinates and manages on-site operations and employees within Terminals 2 and 3
- 33 years of experience at KLM
- 5 years in current role
- Direct operational involvement in the process (on the floor)

Problems / Bottlenecks

- Belly carts are not always positioned correctly in T3, requiring forklift drivers to manually scan and relocate XPS packages individually
- XPS shipments exceeding 23 kg are not always detected in advance, resulting in unnecessary transport to T1 before being redirected back to T3
- Dangerous Goods (DG) shipments sometimes obstruct belly cart staging areas
- Late transfer of belly carts results in missed flights
- During busy periods, T3 employees occasionally forget to transfer belly carts to T1
- Significant time lost due to forklift drivers repeatedly mounting and dismounting forklifts to manually scan and relocate individual packages
- Approximately one-third of night operations consist of XPS shipments, with the remaining two-thirds being general cargo (particularly during weekends)
- XPS packages occupy valuable space in the T3 Voorloods, sometimes with only a few parcels placed on wooden pallets
- In the previous Chain system, multiple packages from a single truck could be processed simultaneously; in iCargo, each package must be scanned, checked, and weighed individually, increasing processing time
- Under Chain, it was possible to identify trucks carrying only XPS in advance and prioritize them; this visibility has been lost in iCargo
- Drivers with only 1–4 small packages may wait 1–2 hours, as workers are not permitted to prioritize them to ensure fairness
- Trucks carrying both XPS and general cargo are sometimes redirected to T3 as early as 21:00, despite T1 being operational until 22:30
- XPS shipments containing Dangerous Goods are occasionally not correctly flagged in the system if documentation is incomplete, resulting in incorrect routing (e.g., missing "go to KI" instruction)

Desirables / Goals

- Remove as many XPS shipments as possible from T3 nighttime operations
- Ensure the system can manage failures and shipments that do not meet specifications
- Provide accessible assistance for drivers in case of issues or questions
- Ensure flagged or non-compliant packages can be set aside for additional checks before system entry

Stakeholders

- Clients / truck drivers
- Warehouse workers

Drop-off Solution

- Viewed as a strong concept if properly implemented
- Expected to benefit both KLM employees and drivers
- Requires drivers to ensure packaging and labeling compliance prior to delivery

Additional Notes

- During daytime at T1, only XPS is unloaded from mixed-cargo trucks; remaining cargo is sent to T3
- Typically 2–3 belly carts are transported at a time from T3 to T1 (maximum of 6 possible)
- On weekends, T1 employees are responsible for retrieving belly carts from T3
- To save time, drivers sometimes scan multiple XPS shipments collectively before transporting them to belly carts
- Upon scanning, iCargo automatically assigns location "DWXPS"
- A single shipment may consist of multiple packages
- Transfer time from T3 to T1 is approximately 5–10 minutes
- Pharma XPS shipments require three additional acceptance checks, which can be efficiently performed by a human operator

D.1.8. Interview 8 of 9

Interviewee Experience

- Coordinates and manages on-site operations and employees within Terminal 1, with direct involvement in the sorter system
- 33 years of experience at KLM
- Approximately 1 year in current role
- Direct operational involvement in the process (on the floor)

Problems / Bottlenecks

- Officially, packages at T3 should be sorted by departure time (e.g., before 12:00 and after 12:00), but this is not consistently implemented
- During Saturday peak operations, up to 10 carts of packages may accumulate
- Saturday represents the busiest operational peak
- Shipment location remains listed as "DWARS" throughout the transfer process until manually scanned at T1
  - Even after insertion into the sorter, location remains "DWARS" until scanned together with the assigned unit
  - In the previous Chain system, chute-level visibility was available
- XPS shipments are often transferred to the Oogplein too late
- Service commitments to clients (including premium express services) are not consistently met
- No prioritization of XPS shipments on belly carts arriving at T1
- Shipments miss early morning departures
- Communication and coordination between T1 and T2/T3 are inconsistent; responsibility is shared
- Work instructions are not consistently followed
- Work instructions are sometimes unclear or misaligned between terminals

Desirables / Goals

- Ensure accurate acceptance checks and clear accept/reject decisions
- Provide a support hotline for drivers in case of technical issues or breakdowns
- Clear and standardized instructions for both drivers and operational staff
- Extensive system testing prior to implementation
- Consider inclusion of a buffer/storage area at T1

  - Temporary storage for early-arriving shipments when sorter capacity or allocation is unavailable

Stakeholders

- Vanderlande (conveyor systems supplier)
- Lödige (sorter supplier)
- Clients
- BPI (Business Process Improvement)

Drop-off Solution

- Considered highly effective if implemented correctly
- Sorter does not need to be operational at night; only conveyors would need to operate
- Sensors would need to be installed along conveyors to enable automated start/stop functionality
  - Without sensors, conveyors would need to run continuously
- Expected improvement in on-time performance
- Greater consolidation of XPS operations under one roof
- Security considerations:
  - If a truck is opened without KLM supervision, shipment security is compromised
  - Secure seal ("zegel") ensures shipment integrity; if broken outside KLM presence, unloading may not be permitted
  - Potential implementation of X-ray screening and camera systems

Additional Notes

- Only limited categories of Dangerous Goods (e.g., certain batteries) can be handled directly at T1
- T1 employees must continuously monitor the "DWARS" location (at T2) while simultaneously transferring XPH and general cargo from T1 to T3
- Organizational hierarchy: Shift Leader → Team Coordinators → Team Members
- Communication follows similar hierarchical structure between T1 and T2/T3
- Offloading process at T1:
  - Driver registers at Documentation
  - Receives token indicating destination terminal
  - If XPS present, token directs driver to T1
  - T1 office is notified
  - Driver docks and unloads XPS only
  - KLM staff perform document checks and scanning
  - If fully compliant, shipment enters sorter and is routed automatically to correct chute
  - Non-compliant shipments (overweight/oversized) are built onto a unit in the T1 Voorloods and transferred to AL-banen
- Official closing time of T1 is 23:00; in practice, trucks are typically redirected to T3 after 22:30
- "DWARS" and "DWXPS" are distinct location codes
- XPH from Europe is handled at T3; XPH from Amsterdam to Europe is handled at T1 (not via sorter)
- Packages cannot be delivered on skids and must meet dimensional constraints
- "Long John" refers to the conveyor connecting the T1 Voorloods to the sorter
- Conveyors and sorter are separate systems; the sorter refers specifically to the rail-based sorting machine

- Documentation only accepts shipments from secure clients, reducing the need for redundant security checks at terminal level

D.1.9. Interview 9 of 9

Interviewee Experience

- Works within compliance and security operations at KLM Cargo
- Extensive experience with Dangerous Goods processes, regulatory compliance checks, and cargo security regulations
- 30 years of experience at KLM
- Approximately 3 years in current role
- Indirect involvement in operations with oversight of floor-level compliance and security processes

Problems / Bottlenecks

- No X-ray machines currently installed on KLM Cargo premises
- Security dog inspections are currently not being performed

Desirables / Goals

- Autonomous systems must meet or exceed the safety and security standards of human checks
- Must comply with all "Ready for Carriage" requirements
- System must be capable of detecting:
  - Leaks
  - Minor physical damage to parcels
- Ability to verify legitimacy of AOG (Aircraft on Ground) shipments
  - Confirm aircraft registration linked to booking
  - Prevent misuse of AOG priority labeling
- Must comply with regulations requiring truck seals to be opened only in the presence of KLM documentation personnel
- X-ray screening would be preferable, provided that non-X-ray-compatible cargo is excluded from the automated process

Stakeholders

- Customs authorities and regulatory bodies

Drop-off Solution

- Recognized potential efficiency improvements
- Skepticism regarding full autonomy in relation to security validation and compliance performance

Additional Notes

- Security regulations for general cargo apply equally to XPS shipments
- XPS is a product category; security standards remain identical
- AOG (Aircraft on Ground):
  - Indicates an aircraft is grounded and awaiting critical parts
  - One of the highest priority shipment labels
  - Occasionally misused to obtain priority handling
  - Typically granted extended acceptance deadlines
- AWB: Air Waybill (physical document)

- FWB: Freight Waybill (digital document)
- KLM Cargo operates within a closed secure supply chain
- All parties are registered and audited by the Koninklijke Marechaussee
- Business-to-business operations only; no individual/private customers
- Truck seals are assumed tamper-evident and must remain intact
- If a truck carries cargo for both T1 and T3:
  - Truck reports to T1 first
  - Seal is broken for T1 offload
  - Truck should be resealed before proceeding to T3 (not always done in practice)
- Automated systems may incorrectly flag keywords (e.g., “Air Force” in product descriptions)
- KLM Cargo currently has no X-ray equipment on site
- Security dogs are currently out of service
- Although X-ray capability is desired, current processes rely on secured supply chain status rather than physical screening

E

Appendix E

E.1. Relevant data

E.1.1. XPS packages per night-shift

Below is a table outlining the number of XPS packages that were accepted during the night-shift per day during the period of data provided by KLM Cargo. The data for Saturday and Sunday are shown in gray to provide context as to why they are excluded from this project due to their low occurrence.

Table E.1: Number of XPS packages processed during night operations per operational weekday and week number

Week	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Total
23	13	29	8	4	23	0	0	77
24	0	10	55	24	78	0	0	167
25	20	79	22	46	83	2	1	253
26	45	31	20	12	48	1	1	158
27	25	26	10	11	21	0	0	93
28	30	7	6	16	27	1	0	87
29	6	32	46	0	103	3	1	191
30	7	15	66	12	70	3	0	173
31	15	16	71	10	14	1	0	127
32	4	10	23	19	43	5	0	104
33	6	25	4	32	5	0	0	72
34	6	15	8	13	23	0	0	65
35	14	14	1	0	1	0	0	30
36	0	2	17	18	13	0	0	50
37	12	11	6	40	0	0	0	69
38	37	11	58	5	64	0	1	176
39	14	10	17	15	33	3	0	92
40	11	13	115	40	16	0	0	195
41	1	17	3	4	28	0	0	53
42	37	13	33	6	3	0	0	92
43	4	27	26	15	20	1	0	93
44	25	47	18	0	27	1	0	118
45	8	32	14	57	40	0	0	151
46	63	42	4	13	33	0	0	155
47	26	7	7	41	66	1	0	148
48	9	39	28	48	47	0	0	171
49	24	30	21	27	35	0	0	137
50	33	32	28	8	48	0	0	149
51	32	20	29	12	82	0	0	175
52	34	31	7	0	0	0	0	72
1	9	3	0	3	5	0	0	20
2	4	11	0	2	1	0	1	19
Total	631	724	771	553	1100	22	5	3806
Average	17.94	22.09	24.09	17.28	34.38	0.69	0.16	116.63

Table E.2: Summary statistics for weekday XPS packages delivered per night (excluding weekends)

Statistic	Value
Minimum	0 packages
Median	16.5 packages
Mean	23.16 packages
Maximum	115 packages
Standard deviation	21.42 packages
Variance	458.93 packages^2

E.1.2. XPS package weights

Below are a distribution table and histogram for the weights of night-time XPS packages being handled during the provided time period.

Table E.3: Distribution table of package weights

Package weight (kg)	Count
(00 – 01]	326
(01 – 02]	439
(02 – 03]	382
(03 – 04]	382
(04 – 05]	362
(05 – 06]	323
(06 – 07]	249
(07 – 08]	229
(08 – 09]	197
(09 – 10]	171
(10 – 11]	114
(11 – 12]	131
(12 – 13]	111
(13 – 14]	71
(14 – 15]	35
(15 – 16]	45
(16 – 17]	27
(17 – 18]	30
(18 – 19]	12
(19 – 20]	26
(20 – 21]	9
(21 – 22]	15
(22 – 23]	11

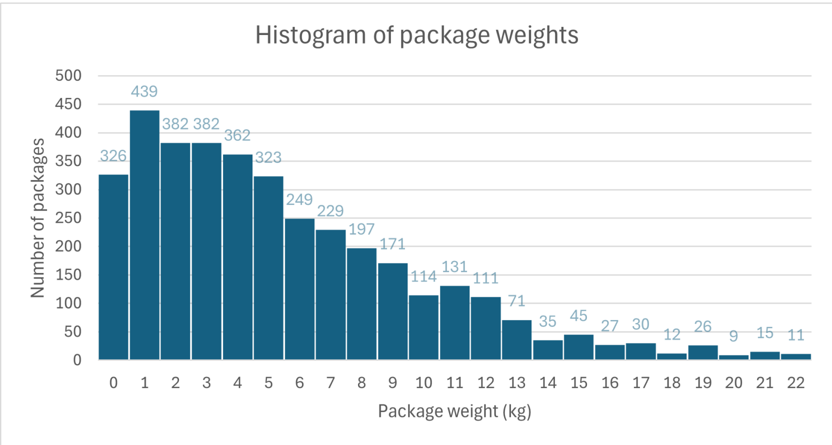


Figure E.1: Histogram of package weights

E.1.3. XPS package volumes

Below are a distribution table and histogram for the volumes of night-time XPS packages being handled during the provided time period.

Table E.4: Distribution table of package volumes

Package volume (m^3)	Count
(0.000 – 0.025]	1783
(0.025 – 0.050]	1092
(0.050 – 0.075]	374
(0.075 – 0.100]	284
(0.100 – 0.125]	62
(0.125 – 0.150]	45
(0.150 – 0.175]	13
(0.175 – 0.200]	18
(0.200 – 0.225]	3
(0.225 – 0.250]	10
(0.250 – 0.275]	2
(0.275 – 0.300]	5
(0.300 – 0.325]	2
(0.325 – 0.350]	2
(0.350 – 0.375]	0
(0.375 – 0.400]	0
(0.400 – 0.425]	0
(0.425 – 0.450]	0
(0.450 – 0.475]	0
(0.475 – 0.500]	1
(0.500 – 0.525]	0
(0.525 – 0.550]	0
(0.550 – 0.575]	0
(0.575 – 0.600]	1

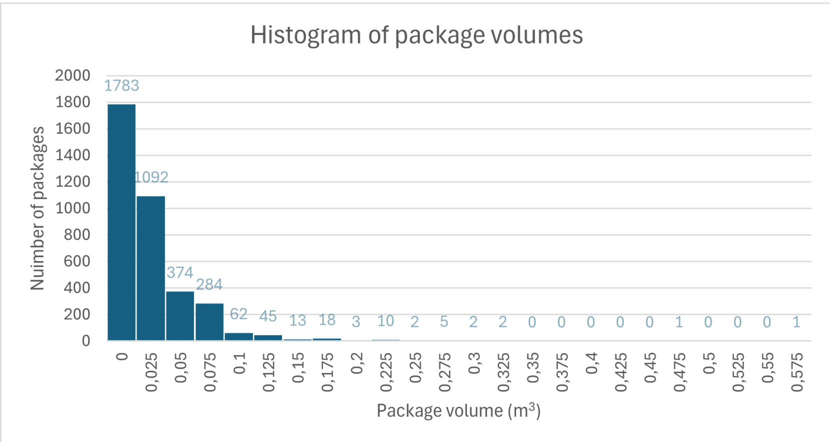


Figure E.2: Histogram of package volumes

E.1.4. Trucks per night-shift

Below is a table outlining the number of trucks carrying XPS packages that were handled during the night-shift per day during the period of data provided by KLM Cargo. The data for Saturday and Sunday are shown in gray to provide context as to why they are excluded from this project due to their low occurrence.

Table E.5: Number of night-time truck arrivals per weekday and week number

Week	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Total
23	4	9	9	7	5	0	0	34
24	0	8	7	8	17	0	0	40
25	15	18	14	10	15	1	0	73
26	11	15	14	9	12	1	0	62
27	11	11	11	6	14	0	0	53
28	11	11	13	9	9	0	0	53
29	8	14	10	8	13	3	1	57
30	3	8	18	11	15	1	0	56
31	8	8	2	10	15	0	0	43
32	8	14	12	8	11	1	1	55
33	7	9	7	10	11	0	0	44
34	6	8	8	4	10	0	0	36
35	8	9	8	4	1	0	0	30
36	2	7	6	9	11	0	0	35
37	8	8	7	10	13	0	0	46
38	8	9	9	8	12	0	0	46
39	8	9	5	6	11	4	0	43
40	7	11	10	8	16	0	0	52
41	6	11	9	6	11	0	0	43
42	12	10	8	10	10	1	0	51
43	9	11	10	7	6	2	0	45
44	10	14	10	6	1	0	0	41
45	6	9	8	6	8	2	0	39
46	11	5	9	7	9	0	0	41
47	5	12	7	11	15	1	0	51
48	4	8	7	5	2	0	0	26
49	8	10	5	11	10	0	0	44
50	9	9	10	4	5	1	0	38
51	3	9	10	9	15	1	1	48
52	9	7	6	0	1	0	0	23
1	4	6	0	1	5	0	0	16
2	3	0	0	0	0	0	0	3
Total	232	307	269	228	309	19	3	1367
Average	19.12	21.94	23.36	16.76	33.33	0.66	0.15	115.33

Table E.6: Summary statistics for weekday truck arrivals (excluding weekends)

Statistic	Value
Minimum	0 trucks
Median	9 trucks
Mean	8.41 trucks
Maximum	18 trucks
Standard deviation	3.84 trucks
Variance	14.71 trucks^2

E.1.5. XPS packages per truck

Below are a distribution table and histogram for the number of night-time XPS packages per truck being handled during the provided time period.

Table E.7: Distribution table of packages per truck

Packages per truck	Count
(00 – 02]	457
(02 – 04]	138
(04 – 06]	65
(06 – 08]	47
(08 – 10]	20
(10 – 12]	12
(12 – 14]	17
(14 – 16]	10
(16 – 18]	6
(18 – 20]	6
(20 – 22]	5
(22 – 24]	2
(24 – 26]	3
(26 – 28]	1
(28 – 30]	3
(30 – 32]	1
(32 – 34]	1
(34 – 36]	1
(36 – 38]	1
(38 – 40]	3
(40 – 42]	2
(42 – 44]	1
(44 – 46]	1
(46 – 48]	2
(48 – 50]	0
(50 – 52]	2
(52 – 54]	0
(54 – 56]	1

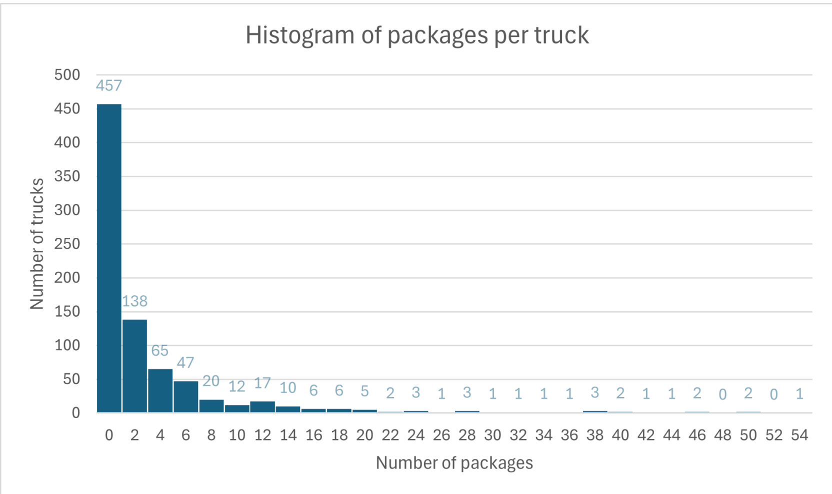


Figure E.3: Histogram of packages per truck

E.1.6. Truck interarrival times during night-shift

Below are a distribution table and histogram for the times between night-time truck arrivals, or *interarrival times*, for the trucks being handled during the provided time period.

Table E.8: Distribution of truck interarrival times

Interarrival time (min)	Count
(000 – 010]	343
(010 – 020]	210
(020 – 030]	171
(030 – 040]	107
(040 – 050]	81
(050 – 060]	63
(060 – 070]	39
(070 – 080]	33
(080 – 090]	22
(090 – 100]	24
(100 – 110]	18
(110 – 120]	12
(120 – 130]	9
(130 – 140]	11
(140 – 150]	7
(150 – 160]	5
(160 – 170]	4
(170 – 180]	3
(180 – 190]	7
(190 – 200]	9
(200 – 210]	2
(210 – 220]	2
(220 – 230]	1
(230 – 240]	4
(240 – 250]	1
(250 – 260]	1
(260 – 270]	0
(270 – 280]	1
(280 – 290]	1
(290 – 300]	0
(300 – 310]	0
(310 – 320]	0
(320 – 330]	1

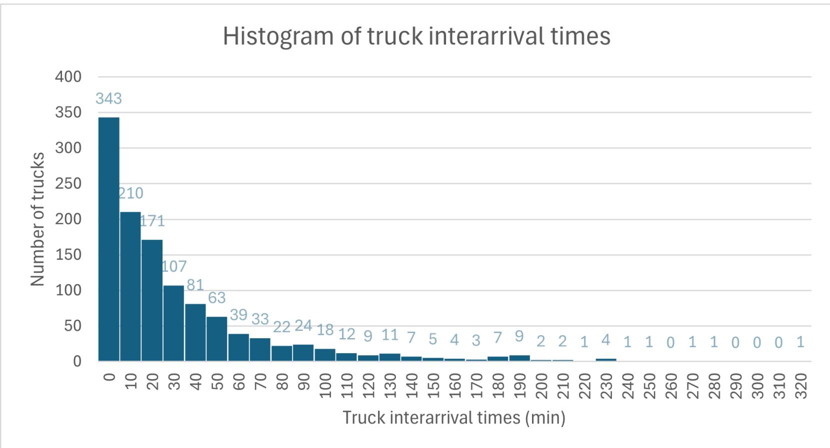


Figure E.4: Histogram of truck interarrival times

E.1.7. Truck waiting times

Below are a distribution table and histogram for the waiting times of trucks being handled at night during the provided time period.

Table E.9: Distribution table of truck waiting times

Waiting time (min)	Count
(000 – 020]	73
(020 – 040]	158
(040 – 060]	202
(060 – 080]	179
(080 – 100]	128
(100 – 120]	115
(120 – 140]	89
(140 – 160]	54
(160 – 180]	62
(180 – 200]	45
(200 – 220]	35
(220 – 240]	29
(240 – 260]	21
(260 – 280]	20
(280 – 300]	11
(300 – 320]	7
(320 – 340]	7
(340 – 360]	4
(360 – 380]	4
(380 – 400]	3
(400 – 420]	1
(420 – 440]	4
(440 – 460]	1

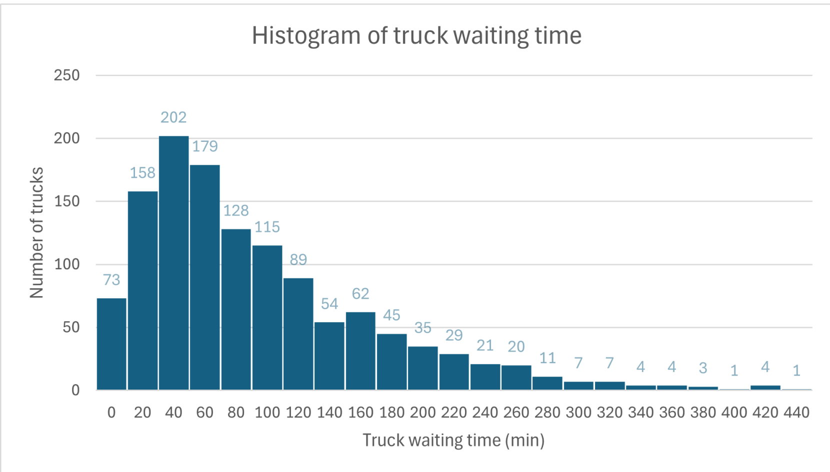


Figure E.5: Histogram of truck waiting times

F

Appendix F

F.1. Simulation run examples

When the model written in code was run and the process simulated, a long list of results were generated. This includes important information from each run iteration as well as vital averaged statistics across all runs.

F.1.1. Current process simulation code results

Note: for the purposes of succinctness in this report, only one of the run iterations will be shown, but in the running of the code, all iterations can be seen. This file represents the current process as if it were unaffected by T3 operations and trucks with only general cargo (i.e. a perfect scenario for the current XPS delivery process)

```


====================================================================
                 CURRENT PROCESS SIMULATION RESULTS
====================================================================


===================================================
RUN 100 / 100
===================================================

GENERATED TRUCK INPUT DATA:
Number of trucks:   9
Interarrival times (min):  40   66   42   51   69   52    6   41    1
Arrival times (min):       40  106  148  199  268  320  326  367  368
Packages per truck:         3    3   10    8    1    1    1   14    1

-----------------------------------------------------------------------------------
1) t_reopen  (Night shift ends):  480.00 min
-----------------------------------------------------------------------------------
Total Trucks arrived during night shift     (T_arrived)           :  9
Total Trucks processed before T1 reopens    (T_beforeT1reopen)    :  9 / 9
Total Trucks still docked and unloading     (T_currentlydocked)   :  0 / 9
Total Trucks still waiting in queue         (T_waitinginqueue)    :  0 / 9

-----------------------------------------------------------------------------------
2) t_lastunload  (Last truck finished unloading):  388.00 min
```

```
-----------------------------------------------------------------------------------
Total Trucks processed          (T_processed)    :  9 trucks
Peak dock queue length:         (T_peak)         :  0 trucks
Average truck waiting time:     (t_Tavg)         :  0.00 min
N95 truck waiting time:         (t_TN95)         :  0.00 min

-----------------------------------------------------------------------------------
3) t_lastXPSscanned  (Last XPS scanned):  411.00 min
-----------------------------------------------------------------------------------
Total XPS packages removed and scanned  (XPS_scanned_total) :  42
Total XPS packages accepted             (XPS_acceptable)    :  42 / 42
Total XPS packages requiring SHC        (XPS_SHC)           :  0 / 42
Total XPS packages not accepted         (XPS_unacceptable)  :  0 / 42

-----------------------------------------------------------------------------------
4) t_totalprocess  (Last XPS sorted):  485.25 min
-----------------------------------------------------------------------------------
Total XPS packages inserted into sorter   (XPS_sorter)   :  42

Time first XPS enters sorter      (t_firstXPSinserted)   :  480.25 min
Time last XPS enters sorter       (t_lastXPSinserted)    :  485.25 min

Max time an XPS package spent in system     (t_XPSmax)   :  440.50 min
Min time an XPS package spent in system     (t_XPSmin)   :  116.50 min
Average time XPS package in system          (t_XPSavg)   :  246.32 min
Median time XPS package in system           (t_XPSmed)   :  283.50 min
Std. dev. XPS package in system             (t_XPSstd)   :  110.70 min
N95 XPS package in system                   (t_XPSN95)   :  436.97 min


====================================================================
      INPUT STATISTICS ACROSS CURRENT PROCESS RUNS (100 RUNS)
====================================================================

TRUCKS PER RUN
Maximum trucks in one run         :  23 (Run 71)
Minimum trucks in one run         :  6 (Run 44)
Median trucks per run             :  13.00
Average trucks per run            :  13.46
Std. dev. trucks per run          :  3.47

PACKAGES PER RUN
Maximum packages in one run       :  241 (Run 88)
Minimum packages in one run       :  15 (Run 78)
Median packages per run           :  57.50
Average packages per run          :  64.88
Std. dev. packages per run        :  38.02

====================================================================
      AVERAGED CURRENT PROCESS RESULTS OVER 100 RUNS
====================================================================

Average trucks arrived                      :  13.46
Average trucks processed before T1 reopen   :  12.96
Average trucks processed total              :  13.42
Average trucks waiting at reopen            :  0.04
```

```
Average trucks docked at reopen           :  0.47

Average last unload time                  :  466.90 min
Average last XPS scanned time             :  480.02 min
Average first XPS inserted into sorter    :  492.93 min
Average last XPS inserted into sorter     :  500.06 min
Average total process time                :  500.06 min

Average truck waiting time                :  0.70 min
Average truck waiting N95                 :  3.71 min

Average XPS scanned total                 :  64.88
Average XPS accepted                      :  58.52
Average XPS SHC                           :  5.05
Average XPS unacceptable                  :  1.31

Average mean XPS flow time                :  248.99 min
Average median XPS flow time              :  241.50 min
Average N95 XPS flow time                 :  430.33 min
Average min XPS flow time                 :  53.53 min
Average max XPS flow time                 :  460.93 min
Average std. dev. XPS flow time           :  118.91 min
```

F.1.2. New process simulation code results

Note: for the purposes of succinctness in this report, only one of the run iterations will be shown, but in the running of the code, all iterations can be seen. In order to obtain the results for the varying percentages, the parameters for the distributions of the XPS packages were changed in the parameters.py file and then the values below would be slightly different for each combination but the overall format and display is identical.

```
====================================================================
                   NEW PROCESS SIMULATION RESULTS
====================================================================


===================================================
RUN 100 / 100
===================================================

GENERATED TRUCK INPUT DATA:
Number of trucks:   12
Interarrival times:   64    39    38    41     9     4    69    19     7   107    11    14
Arrival times:        64   103   141   182   191   195   264   283   290   397   408   422
Packages per truck:    1    10     3     2     3     1     2     4     2     7    39     2

------------------------------------------------------------------------------------

1) t_reopen  (Night shift ends):  480.00 min
------------------------------------------------------------------------------------

Total Trucks arrived during night shift     (T_arrived)             :  12
Total Trucks processed before T1 reopens    (T_beforeT1reopen)      :  11 / 12
Total Trucks still docked and unloading     (T_currentlydocked)     :  1 / 12
Total Trucks still waiting in queue         (T_waitinginqueue)      :  0 / 12

------------------------------------------------------------------------------------

2) t_lastunload  (Last truck finished unloading):  458.00 min
------------------------------------------------------------------------------------

Total Trucks processed                 (T_processed)    :  12 trucks
```

```
Peak drop-off machine queue length  (T_droppeak)    :  1 trucks
Average truck waiting time          (t_Tavg)        :  2.42 min
N95 truck waiting time              (t_TN95)        :  14.35 min

-----------------------------------------------------------------------------------
3) t_lastXPSscanned (Last auto-scan complete, any route) :  545.00 min
-----------------------------------------------------------------------------------
XPS scanned total                       (XPS_total)             :  76
Accepted direct                         (XPS_accepteddirect)    :  47 / 76
Additional check (buffered)             (XPS_additionalcheck)   :  22 / 76
Too early                               (XPS_tooearly)          :  5 / 76
Rejected and sent to T3                 (XPS_rejected)          :  2 / 76

-----------------------------------------------------------------------------------
4) t_totalprocess (XPS inserted into sorter):  545.00 min
-----------------------------------------------------------------------------------
Total XPS packages inserted into sorter (XPS_sorter)      :  69 / 76

XPS inserted into sorter by t = 480      (XPS_by_reopen)   :  51
XPS inserted into sorter after t = 480   (XPS_after_reopen):  25


t_firstdirectXPS      (First direct XPS inserted)         :  126.00 min
t_lastdirectXPS       (Last direct XPS inserted)          :  539.00 min
t_firstadditionalXPS  (First additional-check inserted)   :  483.00 min
t_lastadditionalXPS   (Last additional-check inserted)    :  545.00 min

Direct accepted flow times (truck arrival -> inserted):
  Max (t_dir_max)     :  131.00 min
  Min (t_dir_min)     :  23.00 min
  Avg (t_dir_avg)     :  56.17 min
  Median (t_dir_med)  :  41.00 min
  Std (t_dir_std)     :  34.55 min
  N95 (t_dir_n95)     :  124.10 min

Additional-check flow times (truck arrival -> inserted after reopen):
  Max (t_add_max)     :  419.00 min
  Min (t_add_min)     :  90.00 min
  Avg (t_add_avg)     :  181.23 min
  Median (t_add_med)  :  128.00 min
  Std (t_add_std)     :  102.56 min
  N95 (t_add_n95)     :  378.25 min

All inserted packages (direct + additional-check):
  Max (t_XPSmax)      :  419.00 min
  Min (t_XPSmin)      :  23.00 min
  Avg (t_XPSavg)      :  96.04 min
  Median (t_XPSmed)   :  77.00 min
  Std (t_XPSstd)      :  86.97 min
  N95 (t_XPSN95)      :  296.80 min


====================================================================
        INPUT STATISTICS ACROSS NEW PROCESS RUNS (100 RUNS)
====================================================================

TRUCKS PER RUN
```

```
Maximum trucks in one run          :  23 (Run 43)
Minimum trucks in one run          :  6 (Run 82)
Median trucks per run              :  12.00
Average trucks per run             :  12.80
Std. dev. trucks per run           :  3.50

PACKAGES PER RUN
Maximum packages in one run        :  157 (Run 88)
Minimum packages in one run        :  14 (Run 91)
Median packages per run            :  51.50
Average packages per run           :  60.66
Std. dev. packages per run         :  31.38

====================================================================
        AVERAGED NEW PROCESS RESULTS OVER 100 RUNS
====================================================================

Average trucks arrived                        :  12.80
Average trucks processed before T1 reopen     :  11.76
Average trucks processed total                :  12.65
Average trucks waiting at reopen              :  0.15
Average trucks docked at reopen               :  0.91

Average last unload time                      :  467.41 min
Average last auto-scan time                   :  491.39 min
Average first direct XPS inserted             :  68.88 min
Average last direct XPS inserted              :  486.08 min
Average first additional XPS inserted         :  483.00 min
Average last additional XPS inserted          :  512.30 min
Average total process time                    :  513.82 min

Average truck waiting time                    :  4.34 min
Average truck waiting N95                     :  17.36 min

Average XPS spawned total                     :  60.66
Average XPS scanned total                     :  60.66
Average XPS inserted total                    :  53.29
Average XPS accepted direct                   :  36.15
Average XPS additional check                  :  18.25
Average XPS too early                         :  3.15
Average XPS rejected to T3                    :  3.11

Average XPS inserted by t = 480               :  32.24
Average XPS inserted after t = 480            :  20.19

Average mean direct flow time                 :  45.85 min
Average median direct flow time               :  40.04 min
Average N95 direct flow time                  :  84.37 min
Average min direct flow time                  :  23.00 min
Average max direct flow time                  :  92.30 min
Average std. dev. direct flow time            :  20.69 min

Average mean additional flow time             :  263.05 min
Average median additional flow time           :  253.28 min
Average N95 additional flow time              :  394.15 min
Average min additional flow time              :  123.03 min
Average max additional flow time              :  415.41 min
Average std. dev. additional flow time        :  94.01 min

Average mean total flow time                  :  115.35 min
Average median total flow time                :  58.59 min
```

```
Average N95 total flow time             :  350.07 min
Average min total flow time             :  23.00 min
Average max total flow time             :  415.41 min
Average std. dev. total flow time       :  118.45 min
```