Architecting Internet of Aerospace Things: A System For Tracking Passengers Inside An Airport Terminal

Author
Juan José Garau Luis

Supervisors
Prof. Edward Crawley
Dr. Bruce Cameron

System Architecture Lab
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology

September 2017

"We are drowning in information, while starving for wisdom. The world henceforth will be run by synthesizers, people able to put together the right information at the right time, think critically about it, and make important choices wisely."

Edward O. Wilson

Abstract

Experts predict the number of devices connected to *Internet of Things* networks in 2020 will reach 50 billion, being used in numerous industries. In this Thesis we focus on the aerospace scenario and investigate how and where this technology might be applied to airport mobility and security. We initiate the discussion by presenting the concept of *Internet of Aerospace Things*, IoAT, its main premises and the importance of connectivity in the aerospace domain, which form the basis of the objective of our system.

An airport terminal is an environment where different individuals and companies with diverse interests coincide, and the weak connectivity present between these parties leads to a lack of communication that often creates inefficiencies such as unnecessary waits or poor distribution of resources. We focus our study on the passengers, discussing possible solutions for reducing their traverse time. To accomplish that, we propose providing them with information regarding the people density at different points of the terminal based on tracking technologies.

Next, we select RFID as the concept for our system and study the applicability of this technology for tracking passengers inside an airport terminal. Then, we identify a set of key decisions, such as the physical limits of the system or the specific technology to be deployed, in order to study the different possible architectures. To that end, we develop a simulation model incorporating the most important parts of the system, including: An airport model based on Chicago O'Hare International Airport, a passenger trajectory model, a link budget model, a coverage model, a reader placement algorithm, a communication protocol and the limitations of the available RFID technology.

Based on the developed model, we present the architecture tradespace for our system and analyze the best architectures based on the metrics space. We evaluate the influence and impact each decision has on the tradespace, providing further insights on the system performance and cost. Then, we choose the best architectures and present extended simulation results, showing that our system helps to increase the information on the airport passenger density, thus allowing passengers to take efficient decisions as well as helping the airport authority with security control.

Resum

Els experts prediuen que el nombre de dispositius connectats a xarxes d'*Internet of Things* al 2020 arribarà als cinquanta mil milions, estant presents a nombroses industries. En aquest Treball ens centrem en el context aeroespacial i investiguem com i on aquesta tecnologia pot ser aplicada en la mobilitat i seguretat a un aeroport. Comencem la discussió presentant el concepte d'*Internet of Aerospace Things*, IoAT, les seves premisses més característiques i la importància de la connectivitat en l'àmbit aeroespacial, el qual forma la base de l'objectiu del nostre sistema.

Una terminal d'aeroport és un medi on diferents individuals i companyies amb interessos diversos coincideixen, i la connectivitat dèbil que es presenta entre aquestes parts condueix a una falta de comunicació que crea ineficiències, com esperes innecessàries o una distribució pobra dels recursos. Centrem el nostre estudi en els passatgers, argumentant les solucions possibles per a reduir el seu temps de travessia. Per aconseguir-ho, proposem proporcionar-los informació respecte la densitat de persones a diferents punts de la terminal basant-nos en tecnologies de seguiment.

Llavors, escollim RFID com el concepte del nostre sistema i estudiem l'aplicabilitat d'aquesta tecnologia per al seguiment de passatgers dintre una terminal d'aeroport. Després, identifiquem un conjunt de decisions clau, com els límits físics del nostre sistema o la tecnologia específica a ser instal·lada, per tal d'estudiar les diferents arquitectures possibles. Amb aquest fi, desenvolupem un model de simulació incorporant les parts més importants del sistema, incloent: un model d'aeroport basat en l'aeroport de Chicago O'Hare International, un model per a les trajectòries dels passatgers, un model de balanç d'enllaç, un model de cobertura, un algorisme de col·locació de lectors, un protocol de comunicació i les limitacions de la tecnologia RFID disponible.

Basant-nos en el model desenvolupat, presentem el *tradespace* de les arquitectures del nostre sistema i analitzem quines són les millors arquitectures a partir de l'espai de mètriques. Avaluem la influència i impacte que cada decisió té al *tradespace*, proporcionant més idees sobre el rendiment i el cost del sistema. Llavors, escollim les millors arquitectures i presentem resultats ampliats de la simulació, demostrant que el nostre sistema ajuda a incrementar la informació sobre la densitat de passatgers a l'aeroport, permetent que els passatgers prenguin decisions eficients alhora que ajuda en la gestió de la seguretat a l'autoritat aeroportuària.

Resumen

Los expertos predicen que el número de dispositivos conectados a redes de *Internet of Things* en el 2020 llegará a los cincuenta mil millones, estando presentes en numerosas industrias. En este Trabajo nos centramos en el contexto aeroespacial e investigamos cómo i dónde esta tecnología puede ser aplicada en la movilidad y seguridad en un aeropuerto. Empezamos la discusión presentando el concepto de *Internet of Aerospace Things*, IoAT, sus premisas más características y la importancia de la conectividad en el ámbito aeroespacial, el cual forma la base del objetivo de nuestro sistema.

Una terminal de aeropuerto es un medio donde diferentes individuales y empresas con intereses diversos coinciden, y la conectividad débil que se presenta entre estas partes conduce a una falta de comunicación que crea ineficiencias, como esperas innecesarias o una distribución pobre de los recursos. Centramos nuestro estudio en los pasajeros, argumentando las soluciones posibles para reducir su tiempo de travesía. Para conseguirlo, proponemos proporcionarlos información respecto a la densidad de personas en diferentes puntos de la terminal basándonos en tecnologías de seguimiento.

Luego, elegimos RFID como concepto de nuestro sistema y estudiamos la aplicabilidad de esta tecnología para el seguimiento de pasajeros dentro de una terminal de aeropuerto. Después, identificamos un conjunto de decisiones clave, como los límites físicos de nuestro sistema o la tecnología específica a ser instalada, con tal de estudiar las diferentes arquitecturas posibles. Con este fin, desarrollamos un modelo de simulación incorporando las partes más importantes del sistema, incluyendo: un modelo de aeropuerto basado en el aeropuerto de Chicago O'Hare International, un modelo para las trayectorias de los pasajeros, un modelos de balance de enlace, un modelo de cobertura, un algoritmo de colocación de lectores, un protocolo de comunicación y las limitaciones de la tecnología RFID disponible.

Basándonos en el modelo desarrollado, presentamos el *tradespace* de las arquitecturas de nuestro sistema y analizamos cuáles son las mejores arquitecturas a partir del espacio de métricas. Evaluamos la influencia e impacto que cada decisión tiene en el *tradespace*, proporcionando más ideas sobre el rendimiento y el coste del sistema. Luego, elegimos las mejores arquitecturas y presentamos resultados ampliados de la simulación, demostrando que nuestro sistema ayuda a incrementar la información sobre la densidad de pasajeros en el aeropuerto, permitiendo que los pasajeros puedan tomar decisiones eficientes a la vez que ayuda en la gestión de la seguridad a la autoridad aeroportuaria.

Acknowledgements

First, I would like to sincerely thank Prof. Edward Crawley and Dr. Bruce Cameron for the invaluable guidance and support provided during my stay at MIT. Together, they have introduced me to System Architecture and made me discover a new way of tackling engineering problems. Ed, Bruce, I specially appreciate the trust you have deposited in me since minute zero and all the advice received regarding my professional and personal endeavors. It truly has been an honor to work with you.

I am deeply grateful to Dr. Marc Sánchez for all the counseling he has provided me since I arrived to MIT, even after he earned his PhD and was not obliged to do so. Marc, thanks for helping me get through the most critical parts of this research and sharing your experience with me. I would also like to thank all the academics and professionals with whom I have had chance to discuss my work with during these months. Specially, I would like to mention Dr. Rahul Bhattacharyya, for his useful inputs on RFID, as well as Prof. Vincent Chan and Arman Rezaee, for their help during the development of the model.

One of the key factors that has motivated me during my stay has been the great environment at 33-409. Íñigo, Markus, Nikko, Axel, Aleix, Anne, Johannes, Matt, George, you all have been a source of inspiration and made routine easier to endure. I wish you all succeed in your future projects. I want to thank Íñigo particularly for all the programming tips and laughter doses provided.

I do not want to miss the chance to thank Prof. Eduard Alarcón, my Thesis advisor at TelecomBCN, and and Prof. Miguel Ángel Barja, director of CFIS, for helping me fulfill my dream of studying at MIT. Once I was in Boston, my dear friends Miguel, Marc, Carlos, Francesc, Marta and Reus made me feel like home and have continued to do so until the last day. I really hope to see you achieving your goals in the coming years. Spain@MIT, you have rocked too!

This Thesis marks the end of my undergraduate studies in Barcelona. I would like to thank all the incredible people I have met during these years at Penyafort, TelecomBCN, CFIS and ETSEIB. Specially, a huge shout-out to Aitor and Óscar for selflessly helping me during the hardest parts of my courses and for being people to look up to.

Last but not least, I want to thank the tremendous support received from my family and friends in Mallorca during all these years. Big thanks to my parents, Ana and Simón, for encouraging me to work hard and aim high. Also, thanks to my grandparents, godparents, aunts, uncles, cousins and the rest of my family for cheering me up through phone and video calls. Finally, I am grateful to my closest friends —you know who you are—, specially Albert, Mati and Tomeu, for showing me that 6000 kilometers mean nothing when it comes to friendship.

Contents

List of Figures

List of Tables

Abbreviations

ADS-B	Automatic Dependent Surveillance - Broadcast
AFSA	Advanced Framed Slotted ALOHA
AFTP	Arriving Final Tracking Point, *Decision*
AITP	Arriving Initial Tracking Point, *Decision*
APP	Antenna Placement Problem
APR	Antennas Per Reader, *Decision*
ASK	Amplitude-Shift Keying
BSFA	Basic Framed Slotted ALOHA
C1G2	Class-1 Generation-2
CDMA	Code Division Multiple Access
CIR	Channel Impulse Response
CV	Computer Vision
DFSA	Dynamic Framed Slotted ALOHA
DFTP	Departing Final Tracking Point, *Decision*
DITP	Departing Initial Tracking Point, *Decision*
DSM	Design Structure Matrix
DSS	Decision Support System
EDFSA	Enhanced Dynamic Framed Slotted ALOHA
EIRP	Equivalent Isotropically Radiated Power
EM	Electromagnetic
EPC	Electronic Product Code
FCC	Federal Communications Commission
HF	High Frequency
IC	Integrated Circuit
IoAT	Internet of Aerospace Things
IoT	Internet of Things
ISM	Industrial, Scientific and Medical
ISO	International Standards Organization
LAN	Local Area Network
LF	Low Frequency

LIDAR	Light Detection And Ranging
LOS	Line Of Sight
MAC	Media Access Control
NOAC	Number Of ALOHA Cycles, *Decision*
OCR	Optical Character Recognition
OPM	Object-Process Methodology
ORD	Chicago O'Hare International Airport
OSI	Open Systems Interconnection
PL	Path Loss
RADAR	Radio Detection And Ranging
RFID	Radio Frequency IDentification
RPH	Reader Placement Heuristic, *Decision*
RX	Receiver
S-ALOHA	Slotted ALOHA
SUP	Suspected Unapproved Part
TAP	Tracking Arriving Passengers, *Decision*
TCP	Tracking Connecting Passengers, *Decision*
TDMA	Time Division Multiple Access
TDP	Tracking Departing Passengers, *Decision*
TT	Tag Technology, *Decision*
TX	Transmitter
UHF	Ultra High Frequency
WSN	Wireless Sensor Network

Chapter 1

Introduction

The purpose of this chapter is to define the main guidelines of this Thesis, starting with a short introduction about connectivity, *Internet of Things* and its applicability to the aerospace domain, thus presenting what is known as *Internet of Aerospace Things* (IoAT). Then, baseline examples of IoAT projects are discussed, emphasizing those still in development. We choose one of the topics presented, which focuses on connectivity inside airports and how it can benefit the multiple stakeholders involved. Finally, we develop the main objectives of this Thesis, which combines the concepts of System Architecture with the selected project, and disclose its structure.

1.1 Motivation

One of the key elements in the progress of science and technology in the last decades has been the Internet, a global network that links people from all over the world and allows quasi-instantaneous transmission of information. From researchers sharing important pieces of code through online platforms, to global leaders arranging multiple-party video calls to discuss key decisions, *connectivity* has proved to be one of the most valuable assets of mankind, allowing large-scale exchange of information in terms of distance, quantity and quality.

During this century, the constant improvements in the technology for storing (Kryder's Law [9]), processing (Moore's Law [10]) and transmitting (Nielsen's Law [11]) data have led to the creation of new connectivity concepts, one of them being *Internet of Things* (IoT) [12]. IoT aims to create data networks clustering not only computers or mobile devices but also objects whose main function is not being connected, such as means of transportation or small appliances.

In the words of experts, IoT is the creation of a digital replica of a physical object, providing wireless sensors to capture its instrumental nature and therefore provide a digital *connection* to that object. Combining these wireless sensor networks (WSN) with RFID devices, addressing schemes, data storage and data analysis tools [13], numerous industries have benefited from this powerful technology, such as [14][15]

- *Smart cities.* Traffic control, parking optimization...
- *Health-care.* Assisted living solutions, personalized health-care...
- *Environmental monitoring.* WSN in endangered areas, civil protection...
- *Product management.* Supply chain control, good tracking, warehousing optimization...
- *Security.* Identification systems enhancement, large-scale surveillance...

In all these industries, the arise of IoT projects have changed how important issues are controlled and potential risks prevented, by introducing a real-time feedback and control loop.

There is one industry in particular in which connectivity is crucial for the correct functioning of its services and, at the same time, is one of the most limited in terms of connectivity: The aeronautics industry. Recently, the efforts in the modernization of this industry and combination of the new IoT technologies with the current aircraft and airport infrastructures have led to the creation of the *Internet of Aerospace Things* (IoAT) concept.

Due to the fatal consequences technical errors may lead to during flights, all aircrafts are already immensely connected internally, presenting wide networks of sensors and transceivers, duplicated in some cases [16]. However, connectivity is mostly limited to the aircraft itself and basic communications between pilots and ground stations. IoAT aims to boost the level of connectivity between all parties involved in the aviation activities, thus creating a new and broad network to increase the amount of information shared among all nodes.

Given the gap between the development of the IoT technologies in the different industries, we realize the necessity of leveraging IoAT in order to extend the connectivity benefits to the entire aviation environment. In aviation, a good design is important to avoid future problems, so we recognize the benefits regarding IoAT from a System Architecture point of view would suppose and decide to analyze this topic accordingly.

1.2 General Objectives

With the aim to "connect aviation", first we have to understand the current state and developments of the IoAT network. Given the intricacy of this large network and the amount of applications and projects in the aerospace industry that arise from the introduction of IoT technologies, we must focus on a determined application or smaller network in order to make meaningful contributions.

Still, the complexity of the problem demands caution in the proposal and analysis of solutions. Therefore, we opt for applying System Architecture methodology and translate our problem of leveraging IoAT into the determination of the best architecture for a connected system within the IoAT concept.

Accordingly, the main objectives of this Thesis are the study of the connectivity in the aerospace domain, the conceiving of an IoAT system and the application of System Architecture principles to come up with the best architecture possible for the mentioned system.

1.3 Background

The space of possible application that lie in the IoAT domain is broad and a few of these applications have already been implemented or are in way to be so. In order to be able to focus on a specific problem, it is important to first examine these existing projects and analyze its goals and the way they are defined, so we do not have to start our approach from scratch.

From all the interesting applications that have been envisioned, we have selected five that present different types of increased or enhanced connectivity, shown from Table 1.1 to Table 1.5.

Tracking Passengers' Luggage	
Description	RFID tags are attached to checked luggage at airports in order to optimize the transport and classification operations.
Main goals	The system intends to reduce the number of delayed baggages at airports and provide passengers with means to know the status of their luggage during their journey.
Status	Major airlines have showed interested in the technology and some of them such as *Delta* are currently using the technology in regular flights [17].
Connectivity	Present between passengers, airport luggage transporters and luggage itself.

TABLE 1.1: Main features of the *Tracking Passenger's Luggage* IoT application.

Optimizing Airport Inspections	
Description	This idea consists in the attachment of RFID tags and sensors to the different airplane parts to identify emergent issues during flights.
Main goals	The system communicates detailed information of the status of the affected pieces prior to landing and stores a database of the history of modifications the part has undergone, to provide robustness against the suspected unapproved parts (SUP) problem [15].
Status	*Boeing* has recently launched a software tool called Airplane Health Management [18] that focuses on helping in the optimization of solving incidences that occur during flight with the help of RFID. On the other hand, *GE* is already implementing the parts database, aiming at the complete tracking of the parts since its manufacture [19].
Connectivity	Present between aircrafts and airport technicians.

TABLE 1.2: Main features of the *Optimizing Airport Inspections* IoT application.

Automatic Dependent Surveillance - Broadcast (ADS-B)	
Description	ADS-B is a technology in which an aircraft determines its position via satellite navigation and periodically broadcasts it, enabling to be tracked.
Main goals	It intends to provide data infrastructure for inexpensive and accurate flight tracking. The information is received by air traffic control ground stations as well as nearby aircraft, ensuring all agents have the same information.
Status	*Aireon* is a private company that has partnered with *Iridium Communications* to provide ADS-B tracking to foster new methods for aircraft location [20]. The first satellites were launched in the beginning of 2017.
Connectivity	Present between aircrafts, satellites and ground stations.

TABLE 1.3: Main features of the *Automatic Dependent Surveillance - Broadcast* connectivity application.

Personalized Treatment	
Description	System for taking account of the complete passenger's experience during the whole journey, specially focusing on unsatisfactory events in order to offer personalized treatment in posterior flights.
Main goals	The main objective of this system is creating a tracking tool of the passenger's experience and make each passenger feel more valuable. The important information would be shared with all the agents responsible at each step of the passenger's journey.
Status	*Delta* proposed a software tool called Guest Service Tool to provide flight attendants with information regarding high-value costumers in order to offer personalized treatment [21].
Connectivity	Present between passengers and airlines.

TABLE 1.4: Main features of the *Personalized Treatment* IoT application.

One of the most remarkable aspects of these IoAT projects is how they establish different types of connections within all the agents present in the aeronautics environment. Unfortunately, we can not base our study on more than a project, so we have to decide where we want to focus this research and which kind of connectivity we want to create.

We inspire ourselves in the project described in Table 1.4, which aims to improve the

Connected Turbulence	
Description	Creating a large sensor network in which each aircraft acts as a mobile node, by providing constant information on turbulence events.
Main goals	This system focuses on having complete control of the status and location of turbulence areas, in order to avoid the operational costs and passenger discomfort that may originate due to turbulence.
Status	*The Weather Company* has proposed a product named *Total Turbulence* which aims precisely at this system's goals [22].
Connectivity	Present between aircrafts and flight control stations.

Table 1.5: Main features of the *Connected Turbulence* connectivity application.

passenger's experience during a flight by providing the flight crew with tools to get feedback from all passengers, specially the high-value ones. However, we realize that once the flight crew has information on the passengers, there is little they can do to improve it, given each passenger has an assigned seat and a specific flight benefits purchased beforehand. In other words, although the system does increase the connectivity, it fails to substantially benefit from it.

Instead, we propose changing the context of the system and focus on the passengers' experience before the flight, i.e., their time at the airport. Therefore, we also focus on how we can improve the passengers' experience but regarding an environment where they have more flexibility. In this case, the connectivity is present between passengers and the entire airport agents, being the airport authority, the security control responsible, or the airport's businesses workers. From now on, we will have this goal in mind and try to come up with a system that fulfills it, using the concepts and principles from System Architecture.

1.4 Literature Review

One of the most important elements in this Thesis is System Architecture, its methodology and its principles. In this section we intend to present the key notions of System Architecture and the guidelines that define the role of the system architect. A full list of the System Architecture principles and extended definition of its main concepts can be found in Appendix A and Appendix B, respectively.

1.4.1 System Architecture

System Architecture conforms the basis of a practice that aims at designing and analyzing the architecture of systems, specially in their early stages of development, by identifying and making key decisions that capture the most of the system's performance and cost. Although it is a remarkably interdisciplinary methodology, System Architecture has proved to be specifically useful in aerospace and civil engineering [23], domains in which systems present a strong and high level of complexity. System Architecture is a way of understanding, designing and managing complex systems [24].

System Architecture is intimately related to *system thinking*, which is not thinking systematically but a different thinking approach based on the understanding of the question or problem explicitly as a system. We use *Crawley et al.* definition of a system, being "a set of entities and their relationships, whose functionality is greater than the sum of the individual entities" [25].

Every system presents attributes of form and function, and it exists framed in a certain context [25]

- The *form* of a system represents its structure and its physical embodiment, what the system *is*.
- The *function* of a system is the action or set of actions for which a system exists, what the system *does*.
- The *context* of the system is what surrounds it, the entities or systems that are outside of it.

Function is executed by form and it *emerges* from the union of the formal entities of the system. While function is closely related to the system's *performance*, the system's form will determine its *cost*.

The function of a system can be also understood as a process-operand relation executed by the system. When the system performs its function, there is a pattern of transformation, known as *process*, in a certain property of an object, known as *operand*. To better comprehend this definitions we use an analytical representation of form and function known as Object-Process Methodology (OPM) [26][27], a set of graphic elements such as blocks or links conceived by Professor Dov Dori. Following this methodology, in Figure 1.1 we show the basic definition of form and function concepts in OPM. A further and detailed description of the basic OPM tools can be found in Appendix C.

As an example, let's apply System Architecture in thermodynamics and analyze the form and function of a basic dissipating fin. The form of the the fin would be the fin itself, i.e., the metallic, heat-conductor object that is attached to the surface of the object we are interested in cooling. On the other hand, the function of the system would be "dissipating heat", in which "dissipating" is the process and "heat" the operand. The OPM representation of this example is shown in Figure 1.2.

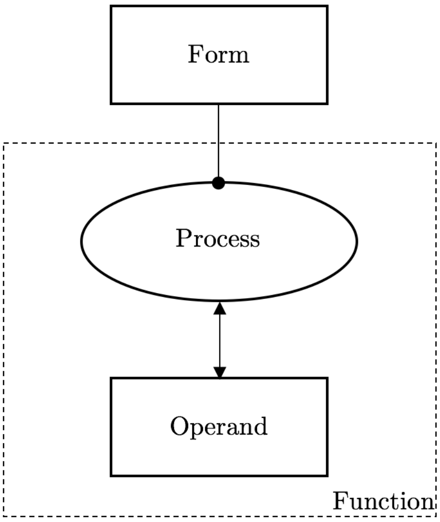


FIGURE 1.1: Generic representation of form and function in OPM.

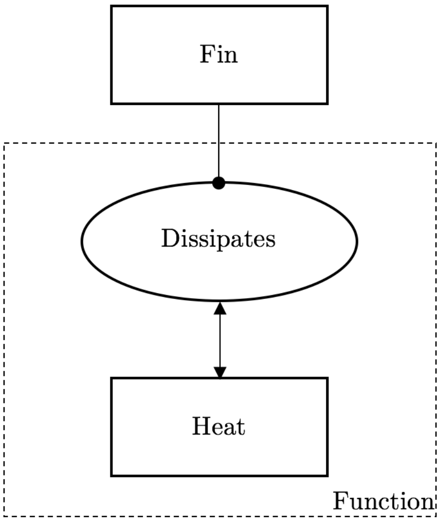


FIGURE 1.2: Representation of form and function for the fin system in OPM.

Once we have presented the basics of system thinking, we are ready to focus on System Architecture and its methodology. System Architecture is defined as "the embodiment of concept, the allocation of physical/informational function to the elements of form, and the definition of relationships among the elements and with the surrounding context" [25].

1.4.2 The Role of the Architect

The *concept* of a system is "a product or system vision, idea, notion, or mental image that maps function to form. It is a scheme for the system and how it works. It embodies a sense of how the system will function and an abstraction of the system form" [25].

The system architect is the responsible of applying the definitions that derive from system thinking to the system in question and make the transition from concept to *architecture*. The architecture is defined as an abstract description of the entities of a system and the relationships between those entities. Both concept and architecture contain mappings of function to form. Having a concept in mind helps in the path to the obtaining of the architecture.

Although it is usual to analyze the architecture of existing or envisioned systems [28], in some cases system architects have neither the concept nor the architecture, and need to come up with the former as a starting point. By analyzing which are the main goals and the needs of the stakeholders involved in the project, the system architect has to propose a solution-neutral function for the system that leads to the selected concept.

According to the Principle of the Solution-Neutral Function [25], the system architect has to avoid being driven by poor system specification or past experience that contain any clues about the system, its form or its function. Instead, a hierarchy of solution-neutral statements should be used to reach the concept of the system.

Sometimes, the space of possible architectures is extensive and it is difficult to analyze and synthesize all possible architectures exhaustively. Instead, the architect has to define a concept without providing too much detail and define the plausible architectures that follow this concept as a set of separated architectural decisions [29][30].

At this point the architect utilizes Decision Support Systems (DSS) [31] to assist in the decision making process. These systems are described by four main tasks or "layers" [25]:

- The *representing* layer includes methods and tools for representing the problem for the human decision maker and encoding the problem for computation. In this layer the architect makes use of tools such as Morphological Matrices [32].
- The *structuring* layer involves reasoning about the structure of the decision problem itself. Design Structure Matrices (DSM) [33] are used to capture the coupling between the different decisions.
- The *simulating* layer is used to determine which combinations of decisions will satisfy logical constraints and calculate the metrics. There is a wide variety of tools that can be used at this point, from simple equations to discrete-event simulation.
- The *viewing* layer presents, in a human-understandable format, decision support information derived from the structuring and simulating layers. Architectural tradespaces, in which the different architectures are compared in terms of performance and cost, are one of the most typical formats [34].

Ideally, after this four step process, the architect is able to determine which architecture will serve best to the goals of the project. However, it is possible different architectures are meet these goals —the non-dominated architectures— and further criteria need to be applied.

1.5 Specific Objectives

In section 1.3 we proposed the definition of a system that increases the connectivity of passengers with the rest of the agents inside an airport terminal, regarding the problem from a System Architecture point of view. After the literature review on System Architecture in section 1.4, we realize we have to conceive an architecture from scratch.

Therefore, the first objective of this Thesis will be the definition of an actual concept that fulfills our goal defined by the stakeholders' needs. To that end, we intend to use the principle of the Solution-Neutral Function.

Following from the concept, we will need to define the architecture as a set of separated individual decisions and therefore an important task will be the construction of the mentioned set. To that end, identifying the elements of form and function —system thinking— will be key to capture the most of the system's performance and cost.

Once we have defined a set of architectural decisions, the four main tasks of DSS will be required to continue with the process. At this point, we will be motivated to use the representing, structuring, simulating and viewing layers to agree upon a specific architecture or group of architectures.

The final goal of this Thesis is to study the viability of the best architectures and discuss the translation of the System Architecture analysis to a real implementation.

1.6 Thesis Overview

Including this introductory chapter, the Thesis is divided in five chapters. In chapter 2 we analyze the problem from the System Architecture point of view, starting from a general goal and narrowing the space of solutions to the RFID specific concept, providing a comprehensive theoretical overview of the technology in order to develop a set of architectural decisions.

Subsequently, chapter 3 covers in detail the simulation model that is developed with purpose of evaluating the different architectures that result from the analysis in the previous chapter. We not only present the distinctness in the modeling of the different architectures but also their shared context and a pair of metrics in order to extract valuable results.

Then, these results are thoroughly discussed in chapter 4 with graphical support from the construction of the architecture tradespace. Also, we carry out further analyses with variation in the input set as well as provide insights on non-architectural but meaningful information.

Finally, chapter 5 concludes this Thesis with a summary of the key contributions and a brief discussion on the future work that follows this research.

Chapter 2

System Architecture

At this point we have an objective to fulfill and an environment in which the connectivity system will have to operate. In this kind of situations, one may tend to base all ideas and possible solutions on previous experiences or similar problems, thus leading the architect to a narrower set of potential options. It is not possible to understand all of system architecture from a reverse engineering perspective. The goal of this chapter is to analyze the problem with the tools and principles of System Architecture. Therefore, we will now shift to an approach of "forward engineering", in order to introduce two important ideas: The solution-neutral function (section 2.1) and the concept (section 2.2) of our system.

In section 2.3 we will present a study on RFID technology from the ISO's OSI model [35] point of view in order to analyze the formal and functional domains of this technology and determine the necessary levels of complexity and abstraction in our system. Based on this study, in section 2.4 we will obtain a set of separate architectural decisions that will lead to the creation of a simulation model to test each one of the feasible architectures.

2.1 Solution-Neutral Function

In order to define the *concept* for our system we first must identify the functional intent expressed as solution-neutral function. We use *intent* to designate goals for a system. To better understand the goals of the system we will focus on the functional intent derived from the primary need of the primary beneficiary. According to the procedure for deriving a solution-neutral statement of a functional intent in [25], we have to consider the following steps:

- Consider the beneficiary.
- Identify the need of the beneficiary we are trying to fill.
- Identify the *solution-neutral operand* that, if acted upon, will yield the desired benefit.
- Identify the *attribute* of the solution-neutral operand that, if changed, will yield the desired benefit.
- Perhaps identify *other relevant attributes* of the solution-neutral operand that are important to the statement and fulfillment of the goal.
- Define the *solution-neutral process* that changes the benefit-related attribute.
- Perhaps identify *relevant attributes* of the solution-neutral process.

In [6] the perspective of the passengers, the airlines, the owners, businesses, and the government when designing an airport terminal is discussed. Each of these stakeholders has its preferences and needs —more than a single one in some cases— and that occurs when introducing a new system inside the airport environment too. Therefore, it is necessary to define a primary beneficiary and focus the development of the solution on it. As the main users of airports' transportation services, *we consider the passengers as the primary beneficiaries of our system.*

Travellers are mainly interested in having a fast transportation from the origin airport to their respective destinations, and that includes the mandatory steps they have to follow before getting on the plane, such as baggage check-in or security controls. But we can also consider that passengers want this transportation to be as secure as possible, introducing a second need. With this in mind, in Table 2.1 we have the following two possible formulations of the solution-neutral functional intent.

Question	**Fast Transportation**	**Security**
Beneficiary?	People	People
Need?	"Save time"	"Travel securely"
Solution-neutral operand?	Passenger	Passenger
Benefit-related attribute?	Transit/Traverse Time	Level of security
Solution-neutral process?	Reducing	Increasing
Attributes of process?	Optimally, fairly and non-disruptively	Equally

TABLE 2.1: Formulations of the solution-neutral functional intent considering the needs of the passenger.

Other formulations, such as fulfilling the needs of the businesses or advertisement companies as in Table 2.2 or prioritizing the airport authority as in Table 2.3, can also be considered.

Question	Businesses	Advertisement Companies
Beneficiary?	People	People
Need?	“Earn money”	“Reach people”
Solution-neutral operand?	Business	Advertisement Company
Benefit-related attribute?	Revenue	Amount of Market Opportunities
Solution-neutral process?	Increasing	Increasing

Table 2.2: Formulations of the solution-neutral functional intent considering the needs of the businesses and advertisement companies.

Question	Airport Authority
Beneficiary?	People
Need?	“Guarantee security”
Solution-neutral operand?	Airport Authority
Benefit-related attribute?	Security Control and Management
Solution-neutral process?	Increasing

Table 2.3: Formulations of the solution-neutral functional intent considering the needs of the airport authority.

Once we have answered the different questions for each of the cases we have considered what we have is a solution-neutral framing of the desired function of the system for each of these cases. Each of these set of answers can be summarized with its desired function represented in OPM (see Appendix C), as in Figure 2.1.

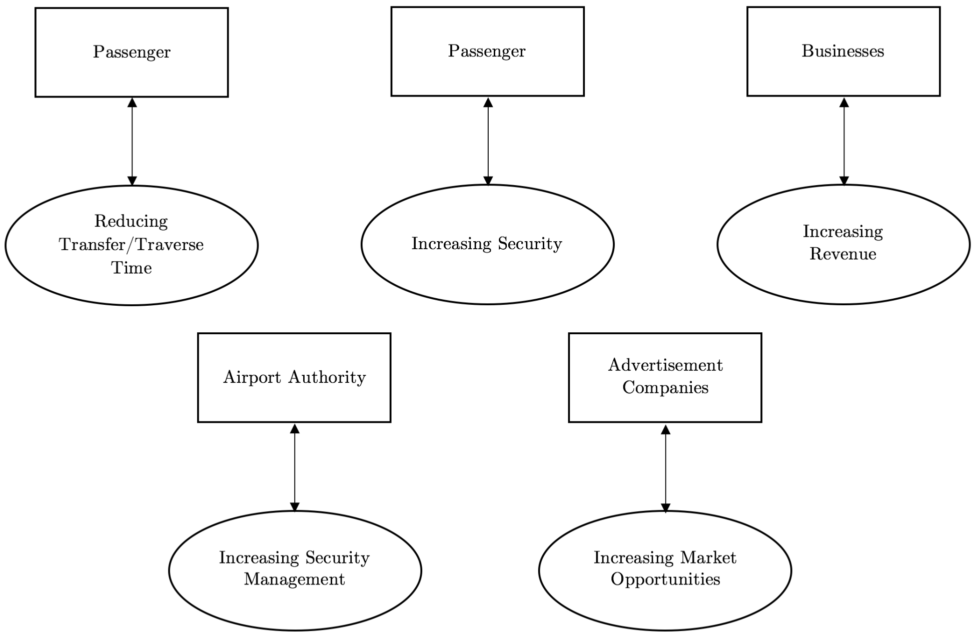


Figure 2.1: Set of possible formulations of the solution-neutral functional intent in OPM.

This diagram contains just a part of all the possible needs of all the stakeholders involved. In order to continue with the selection of the concept we have to set a priority ordering between stakeholders so we can select our solution-neutral function. Our goal is to come up with a system from which the solution-neutral function emerges while satisfying as many of the stakeholders' needs as possible (Principle of Balance, Appendix A).

We will select the solution-neutral function accordingly to the Principle of Value and Architecture (Appendix A), focusing on the stakeholder that perceives the greatest value from an increase of connectivity inside the terminal. Considering value is benefit at cost, *we choose reducing passengers' transfer/traverse time as the solution-neutral function.* This way, from now on, we will prioritize passengers' fast transportation need when developing the concept of our system. Then, as far as possible, we will try to fulfill the security needs of both the passengers and the airport authority.

2.2 Selected Concept

The intellectual distance from the solution-neutral function to the architecture of the system is a large gap to jump. In order to help, we create an intellectual construct known as concept. Concept is not a product/system attribute but a notional mapping between two attributes: Form and function (see Appendix B) [25]. In this section we have to manage to go from the solution-neutral function to a solution-specific concept. It is key to understand that there is not a unique solution-specific concept but a group of them.

Our solution-neutral function is reducing passengers' transfer/traverse time inside an airport. This is the time the passenger takes from getting to one place of the terminal to another location, such as going through the security control or going from the shopping area to the designated gate. In these displacements, passengers encounter with unwanted obstacles that may increase their transfer time. These obstacles can be avoidable, if there are multiple routes, or unavoidable, such as waiting queues. We look for concepts that cut down the impact of the obstacles in the transfer time, either reducing the time spent dealing with them or proposing alternative routes for avoiding them. This way, potential concepts could focus on queue optimization, resource location, or even route suggestion.

Let's start by considering the current resources or locations of an airport terminal, i.e. check-in counters, security points, coffee shops, stores... Increasing the number of resources would also increase the number of passengers served simultaneously and therefore reduce the waiting time in queue. Given the typical distribution of arriving-by-land passengers in airports suggests considering a Poisson model [6], this time reduction is reflected in the expression for the waiting time in a $M/M/c$ queue [36]

$$T_w = \frac{C(c, \lambda/\mu)}{c\mu - \lambda} \tag{2.1}$$

where T_w, $C(c, \lambda/\mu)$, c, λ, and μ stand for the expected waiting time, the erlang-C function, the number of servers, the arrival rate, and the server rate, respectively. Having the number of servers in the denominator makes it have an inversely proportional relationship with the expected waiting time.

Measures such as opening more checkpoints —check-in counters, security points...— or hiring more workers would be specializations of the *increasing checkpoints* concepts. We can see these measures implemented in Brussels airport's new Connector Building [37] or Austin airport's new South Terminal [38]. For instance, in the case of Brussels, the new building includes a central platform for access and security screening as well as border control for travellers departing Pier A. This considerably increases the efficiency of the screening process. Moreover, the building offers plenty of space for a commercial area.

The increase in the number of checkpoints or locations inside an airport has to be accompanied by a change in the physical space of the terminal. Sometimes this change is not possible and therefore concepts dealing with the improvement of the current infrastructure have to be considered. In this second line of concepts lie projects such as UK's attempts on implementing e-borders [39]. The project consisted in the creation of a database storing all travellers information in order to accelerate the border procedural checking and easily detect criminals. Due to disagreements between the interested parties the project was cancelled.

The concepts we have introduced so far focus on specific locations and suggest management of the terminal resources from the airport authority point of view. Although these measures have an impact on the passengers' transfer time they do not contemplate the complete displacement of a passenger inside an airport but just a part of it (just the security control, just the baggage check-in...). Imagine the case a passenger waiting at the designated gate wanted something to eat, where should that passenger go in order to satisfy his/her needs without spending too much time?

Note "too much time" is not a defined quantity, as each passenger has a personal feeling towards the time spent. We should therefore come up with a concept that allowed the passenger bypass avoidable waits independently of the type of the displacement and the passenger itself. Again, as valuing time presents depends on the person who values, we have to think of something that provides the passengers all the necessary tools to value the time, and then let them decide. This suggests gathering data of the current state of the airport at the moment of the passenger's decision on which route to take. This data would not only contain the physical location of the place/s where a passenger can satisfy his/her needs, but also measurements of time based on other passengers' behaviours and previous displacements. Then, the passenger could decide freely which route to take based on suggestions on the shortest-time routes.

The information could be provided to the passenger through an app as they do in Miami International airport [40], or could be displayed in interactive panels as the interactive maps in Boston Logan airport [41]. Providing the spatial information of the terminal is easy, however, gathering the data from the temporal state —passenger density, queue timing...— of the terminal requires more complexity. This can not be done without tracking the passengers inside the terminal.

If the system acquired accurate data of the passengers' trajectories, it could provide valuable and precise information on waiting times and new trajectory durations throughout all the terminal. If we combine this temporal information with the spatial information of the airport —airport map, businesses' location...— we can provide not only passive data but also active data in the form of route or location suggestions that will reduce passengers' transfer time. Therefore, *we introduce tracking passengers as a solution-specific concept for the solution-neutral function of reducing passengers' transfer/traverse time.*

Tracking a person would be the function of our system, we still need to figure out the formal part —which we are about to do—. There exist several technologies that allow the tracking of people. The first possibility we consider is *RF tracking* by using RFID, providing each passenger with a tag —active or passive— and track them with RFID readers to study their trajectory, as in [42][43].

Another tracking method is *light tracking*, which takes the form of a LIDAR. Following this concept, in [44] a system using laser for tracking people inside a museum is discussed. LIDAR is not a popular method for tracking people but for detecting them, as showed in [45]. A close technology to LIDAR is RADAR, which lies in the *radio tracking* concepts, and is also used for tracking people in open spaces [46].

The tracking concepts presented so far do not contemplate that people nowadays carry a device able to connect to networks wherever they go: The mobile phone. *WiFi Tracking* is not something new, in Brussels airport detection systems via WiFi and Bluetooth connections are already implemented [47]. But WiFi tracking is not limited to the use of mobile phones, in [48] alternative WiFi devices are attached to workers in a construction site.

In the era of Artificial Intelligence, computers are now able to "see" and detect objects and persons in images, what is known as Computer Vision. Therefore we can contemplate CV or *image tracking* as a tracking concept. In [49] a system to track multiple occluding people using cameras is presented. In addition to the occlusion, the cameras have to be able to successfully track in the dense dynamic environment that an airport is. This has not been studied but for specific scenarios in small areas [50].

We have examined a wide variety of solution-specific concepts that would serve to our need. All these concepts are summarized and presented in OPM in Figure 2.2. Now it is time to make a decision and choose our final concept in order to proceed with the creation of our system.

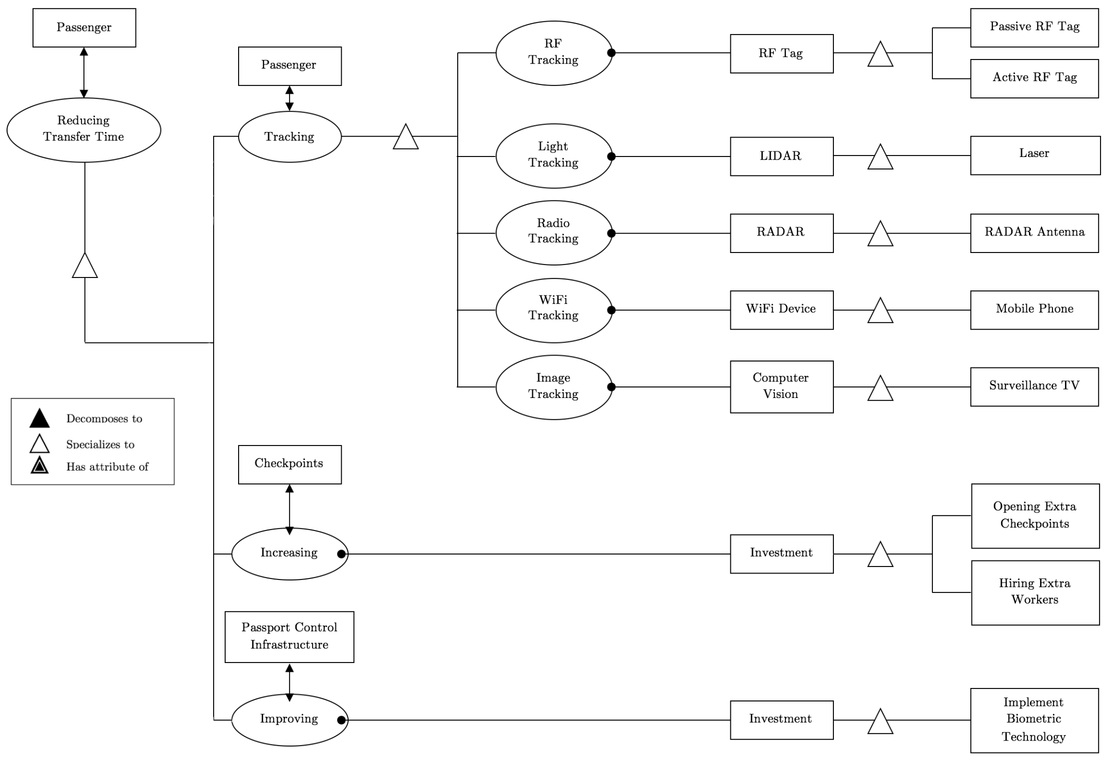


FIGURE 2.2: Solution-neutral function and solution-specific concepts options for the "connectivity" system.

First, we focus on the *Passenger tracking* concepts, as the other concepts do not involve connectivity. From the five different tracking concepts we can observe three of them not only are able to detect and track, but also allow identification. In section 2.1 we introduced the priority ordering between the needs of the stakeholders, and stated that we would also consider *security* as a need as long as we fulfilled the *fast transportation* goals. These three concepts, RFID Tag, WiFi Device and CV, are able to track persons and also identify them, a useful tool for the airport security.

Then, we can observe CV presents a clear disadvantage in front of RFID and WiFi, as the latter two are robust to the physical look of the passengers and do not need a direct path of vision with them. CV could be useful at specific locations such as the security checkpoint [51], but it is non-viable for monitoring an entire terminal.

Between the two final options, WiFi and RFID, *we choose RFID as the solution-specific concept for our system.* This decision is motivated by the non-dependability on each passenger's mobile phone in order to perform the tracking operations. Providing an

RFID tag to each passenger ensures the correct tracking of each individual, avoiding problems such as the disconnection or lack of mobile phones.

In Figure 2.3, the selected concept in OPM is shown, differentiating both its formal and functional domains (see Appendix B). We can appreciate the tracking function for the RFID case needs two devices, an RFID tag for each passenger and several RFID readers located across the terminal.

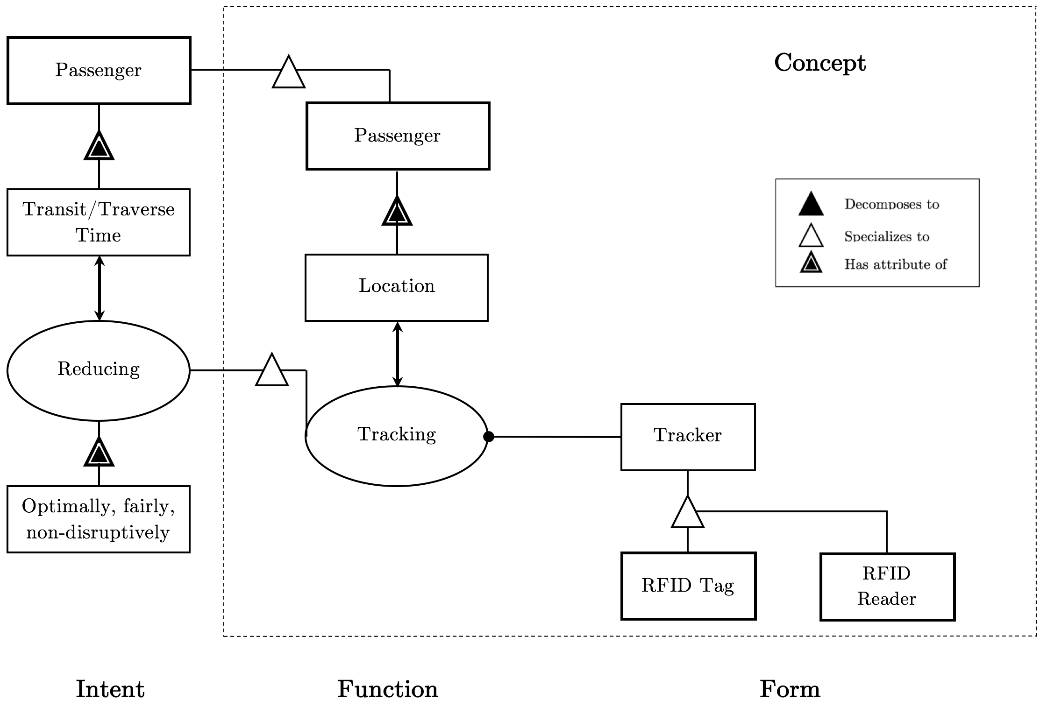


FIGURE 2.3: Selected concept for the "connectivity" system.

The formulation of the specific function and form defining the concept of our system can be found in Table 2.4, which answers the basic questions for defining a concept described in [25].

Question	**Connectivity System**
Specific operand?	Passenger
Benefit-related attribute?	Transit/Traverse Time
Specific process?	Tracking
Attribute of process?	Reliably
Generic concept form?	Tracker
Specific form?	RFID Tag
Attributes of form?	Passive/Active

TABLE 2.4: Formulation of the specific function and form defining the concept of the "connectivity" system.

RFID was created as an automatic identification procedure, with the purpose of providing what are called RFID tags to inanimate objects where identification was necessary.

Given its broad potential, RFID became popular in numerous industries such as retail [52], supply chain [53] or logistics [54]. Then, the extension of RFID identification to RFID tracking, by increasing the number of RFID readers, was natural.

RFID systems are a big component of IoT, as they are usually connected to bigger networks, that allow the remote access to the status and monitoring of tagged objects. Combining both the capacity of information gathering and the "connectivity" of RFID systems we have a powerful tool to ease the controlling, accessibility, and operations of numerous industries.

The increase in the number of readers allows not only tracking [55] but also other functions such as localization [56][57], where one is interested in the exact coordinates of a tagged object rather than an approximate location. The environment created for the tracking and localizing functions is called a *Dense RFID Reader Environment*, which presents specific restrictions regarding the positioning of the antennas. These are discussed in [7][58]. An airport terminal using RFID to track passengers fits perfectly in the Dense RFID Reader Environment description.

In literature, one can find studies on RFID systems used for tracking inside airports. In [59] an RFID-based tracking system for airport security is presented, focusing on the software and hardware architecture of the Tag-Reader communication. It also contemplates how the different tag technologies affect the communication and assesses its operation under real airport conditions (at Athens International Airport). The research proved it is feasible to use RFID cells in airports to detect RFID tags in the far-field range.

In [60] the applicability and advantages of using an RFID system for tracking passengers inside an airport are discussed. It presents the merging between the airport context and the RFID technology, without going deep into any technical detail.

Finally, in [61] an RFID model designed to be used in airports for security and efficiency is presented. It contemplates a cellular network of passive RFID receivers and far-field active RFID tags. It also focuses on the communication problems that may arise such as non line-of-sight conditions or multipath.

In industry, one tech company called *Optag* tried to accomplish our task some years ago [62][63] but unfortunately failed shortly after. Also, in Manchester airport, a similar failing project was proposed at the same time [64]. One reason of this failure in building a solid system might be the lack of research in tracking with RFID that had been done by the time companies realized its broad potential.

From this short literature review we can draw the following conclusions:

- In Academia, there is not much work done in tracking with RFID inside airports.
- In industry, the projects aiming to do so have failed.
- The RFID technology is applicable to the airport environment.
- Most of the research conducted focuses on a single reader.
- For the airport case, there is no research on the dense RFID reader environment.

Given this list of conclusions, in this Thesis we will focus on the implementation of the RFID technology for tracking inside an airport terminal, taking advantage of the previous research on single-reader applications, considering the dense RFID reader environment, and developing the study from a System Architecture point of view. Our next goal is to come up with a set of decisions that capture the essence of the problem and develop a model to study the different architectures.

2.3 RFID Technology

At this point, in order to develop a set of decisions that define our system, it is necessary to study and examine how RFID works and the available RFID technology.

RFID, standing for Radio Frequency IDentification, is one of the automatic identification procedures (Barcode, OCR, Biometrics...) based on the use of electromagnetic waves for the power supply and data-exchange with a data-carrying device [3].

As introduced in previous sections, RFID systems consist of RFID readers and RFID tags. Tags are attached to objects, allowing wireless communication with the readers, which are able to detect the presence of the tags from short to relatively long ranges. This simple mechanism is what makes RFID so popular in numerous applications.

There exist two types of tags: Active and passive. The former contains a radio transceiver and its own power source, while the latter lacks a battery and therefore its data exchange method consists on electromagnetic challenge-and-response coupling. Due to the internal battery, active tags are more expensive and therefore not suitable for large-scale deployment, such as in our airport environment. If each passenger needs a tag, *the best option is to use passive RFID tags*, as they are relatively inexpensive and affordable against a high demand.

A passive RFID tag has three parts: An antenna, a semiconductor integrated circuit coupled to the antenna, and some sort of encapsulation. The powering of the tag and the communication is responsibility of the RFID reader [65].

In the following subsections we will focus on the communication model of the passive RFID tag with the RFID reader. To do so, we will analyze it considering ISO's OSI model [35], specifically paying attention to the Physical and Data Link layers, the first and second layers of the OSI model, respectively.

2.3.1 Physical Layer

In a communication system, the physical layer grants the access to the physical medium, with the settlement and maintenance of physical connections —e.g., antennas, circuits— between the communicating agents from the mechanical, electrical and functional point of view [66].

From this point of view, RFID system can be decomposed into three simple parts: Reader, tag, and a wireless channel, as in Figure 2.4. Transmitter (TX) and receiver (RX) in Figure 2.4 represent the front end of the reader, as readers also have processing and powering units.

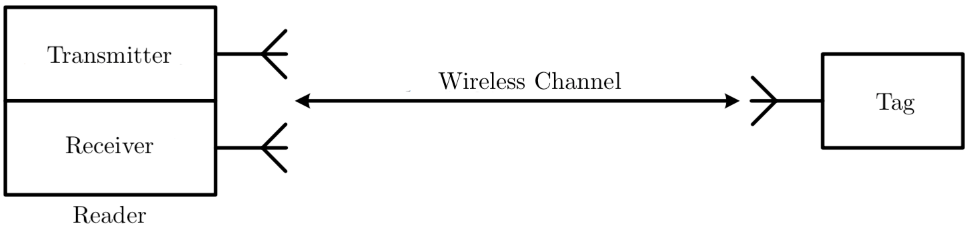


FIGURE 2.4: General building blocks of RFID [1].

In the first stage of the communication process, the TX is responsible for the EM coupling with the tag. Once the tag has been reached and therefore power-supplied, the tag responds and transfers its ID (the tag's chip coordinates this process), which is received by the reader's RX. During the whole process, the encapsulation protects the antenna and chip from environmental conditions or reagents and keeps the tag's integrity [65].

2.3.1.1 Near Field vs. Far Field

Magnetic induction and EM wave capture are the two different RFID design approaches that exist for transferring power from the reader to the tag. Both take advantage of the EM properties associated with an RF antenna — the *near field* and the *far field* [65].

The Near Field system is based on inductive coupling between the tag's antenna and the reader's antenna due to the reactive energy circulating around both. This mechanism obeys Faraday's principle of magnetic induction (Figure 2.5). A magnetic field is generated around the coil of a reader produced by a current flowing through it. As a consequence, the tag's coil reacts to the magnetic field and generates a small current. Then, the effective communication between a reader and a tag is enabled through a mechanism called load modulation [2]. The physical size of the near field RFID tag antenna has little dependence on the frequency used, it is basically a coil [67].

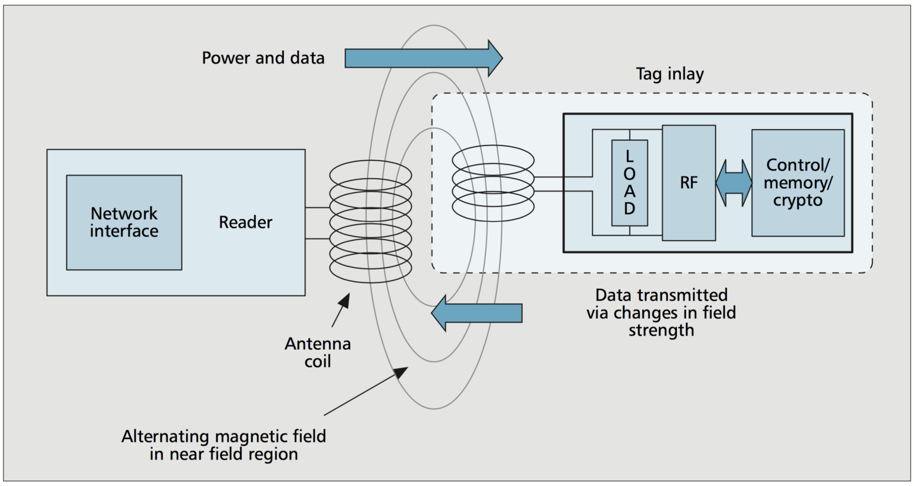


FIGURE 2.5: Near Field communication using inductive coupling [2].

In the near field coupling, the most used frequencies lie in the LF and HF bands, being 128 kHz and 13.56 MHz respectively. This is due to the fact that the boundary between near field and far field regions has an inversely proportional relationship with the carrier frequency of the communication, it is approximately equal to $c/2\pi f$, where c is the speed of light [68]. Therefore, low carrier frequencies predominate in this type of RFID communication. A notorious feature of near field systems is the power drop, which has a rate of $1/r^6$, where r is the distance between a reader and a tag. Another downside is the low bandwidth and, hence, the low data rate [2].

On the other hand, far field systems are based on the free space propagation of EM plane waves. The communication process involves the use of a technique known as backscattering, which consists in the reflection of part of the incident energy on the tag's antenna due to an impedance mismatch between the antenna and the load circuit. The variation in the impedance is what creates the differences in the amount of reflected energy, allowing an effective communication. Figure 2.6 illustrates the mechanism [2]. In this case, the size of the antenna depends on the carrier frequency, as the antenna of the tag is usually a dipole that is half a wavelength long. The relationship is inversely proportional [67].

Regarding the power drop in the far field case, the attenuation is perceived accordingly with a factor of $1/r^2$, which is smaller than the near field case (see Figure 2.7). The

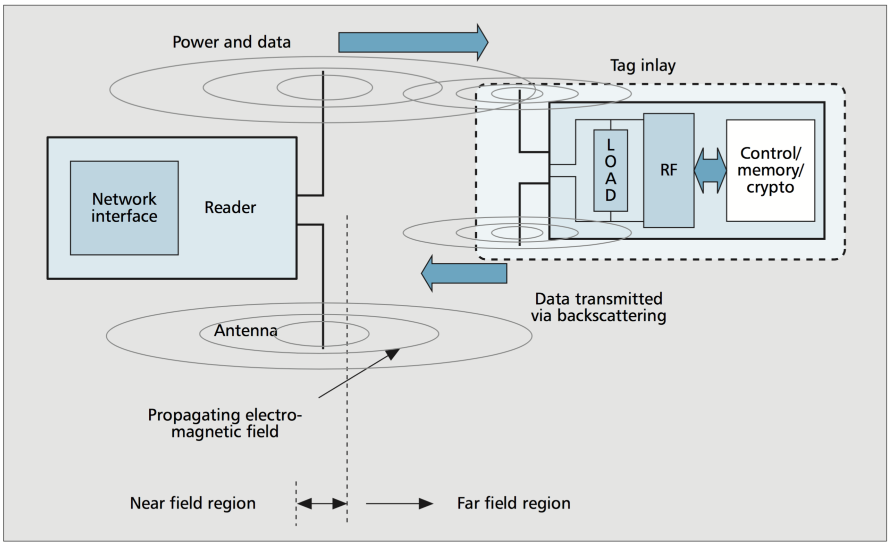


FIGURE 2.6: Far Field communication via backscattering [2].

range achieved by this type of communications lie in the interval 5-20 m, which can be considered as long-range RFID communication. On the other hand, the carrier frequency is higher than in the near field case, as far field tags commonly operate in the 860-960 MHz UHF band, and reaching the 2.45 GHz Microwave band in some cases [2].

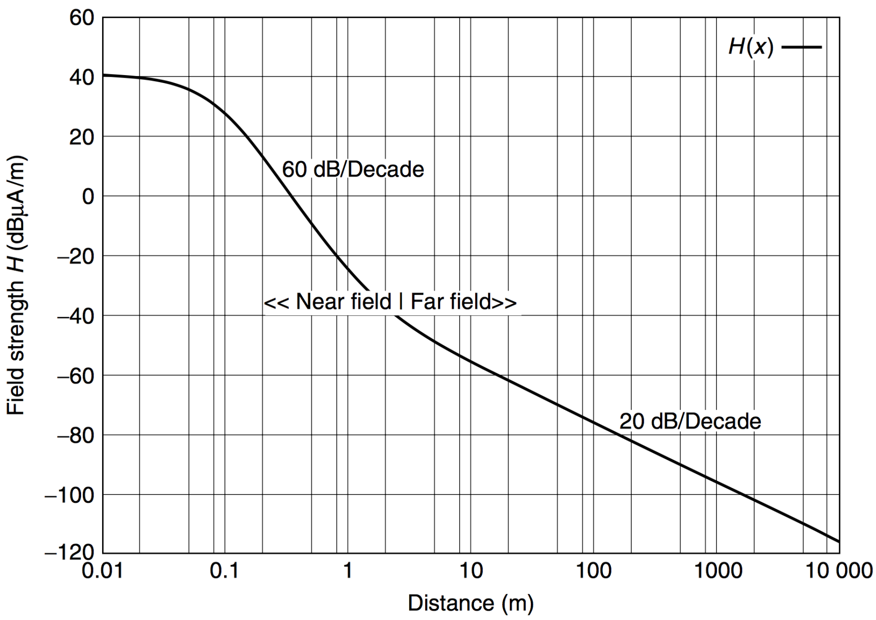


FIGURE 2.7: Graph of the magnetic field strength H in the transition from near to far field at a frequency of 13.56 MHz [3].

In the near-field case, the interest lies in using low frequencies, as the boundary between near-field and far-field shows at greater distances thus systems can rely entirely on magnetic coupling. As an example, for the near-field frequency values given earlier — 128 kHz and 13.56 MHz—, the boundary is located at 370 and 3.5 meters from the reader, respectively. On the other hand, in far-field communication, the closer the boundary is to the reader, the impact proportional to $1/r^6$ is smaller, making far-field RFID applications have a strong interest in using high frequencies. For the 900 MHz and 2.45 GHz cases, the boundary is located at 5 and 2 centimeters from the reader, respectively.

Given the inverse relationship between the tag's antenna's size and the carrier frequency, operating at higher frequencies involves having small antennas, which turn into low fabrication and assembly costs. Then, the current advances in silicon technology combined with the innovative circuit designs have reduced the power consumption of the far field passive tags to only a few microwatts [2].

Considering all the exposed, our system will make use of the far field approach, as proposing a near field communication is non-viable if we want to track all passengers at a reasonable frequency. Our intent will be providing each passenger with a passive, far field RFID tag and the detection —and therefore tracking— will occur when the passenger is in the range of an RFID reader. Then, the terminal will have multiple RFID readers spread.

2.3.1.2 Downlink and Uplink Communication

Regarding the wireless channel we already know two communication links establish through it: A downlink communication in which the reader transmits a modulated RF signal to the tag, and an uplink communication in which the tag's IC varies the input impedance and responds modulating the signal using a mechanism called backscattering. Modulation type often used in RFID is amplitude shift keying (ASK). In ASK modulation, different digital data is transmitted by varying the amplitude of the carrier wave [69].

For the downlink communication, the equation expressing the power P_{rt} received by the tag is [70]

$$P_{rt} = P_{tr} G_{tr} \, PL \, g_{rt} \chi \tau \tag{2.2}$$

Where $P_{tr}G_{tr}$ is the reader transmitted EIRP, g_{rt} is the gain of the tag's receiver antenna, χ is the polarization matching coefficient, and τ is the impedance match between the antenna and the RFID chip. PL represents Path Loss, which strongly depends on propagation environment. If we considered a *Free Space Propagation Model*, Path Loss would be expressed as [71]

$$PL = \left(\frac{\lambda}{4\pi d}\right)^2 \tag{2.3}$$

Where λ is the signal's wavelength, and d is the distance between the reader and the tag. As airports are enclosed spaces, we could also consider the *Indoor Large Scale Propagation Model*, in which the equation for Path Loss is [71]

$$PL(d) = PL(d_0) + 10n\log\left(\frac{d}{d_0}\right) + X_\sigma(\text{dB}) \tag{2.4}$$

Where d_0 is an arbitrary reference distance —usually 1m—, $PL(d_0)$ is the free space path loss for distance d_0, n is the path loss exponent, and X_σ is zero mean Gaussian random variable with variance σ_{dB}^2 in dB. Environmental factors such as obstruction, multipath, tag orientation or displacing objects between TX and RX can affect the signal propagation. To model this random nature of indoor propagation the variable X_σ, called shadow fading, is included [71].

On the other hand, regarding the uplink communication, the power P_{rr} of the backscattered signal modulated by the tag and received by the reader can be calculated as [70]

$$P_{rr} = P_{tr}G_{tr}G_{tt}(PL)^2\Delta\sigma \tag{2.5}$$

Where G_{tt} is the gain of the tag's transmitting antenna and $\Delta\sigma$ is the differential radar cross-section of the tag. This quantity can also be defined as $(g_{rt}\Gamma g_{rr})$, being g_{rt}, Γ, and g_{rt}, the gain of the tag's receiving antenna, the reflection coefficient of the tag, and the gain of the reader's receiver antenna, respectively. In [70] is showed that the limitation of the system is focused on the downlink communication, as the power sensitivity threshold of the tag is more restrictive than the one of the reader.

2.3.1.3 Multipath Propagation

When a TX and a RX communicate through a wireless channel, there is a key property that characterizes this channel and its time-variant channel impulse response (CIR): Multipath propagation. This type of propagation causes the creation of countless paths that define the CIR. On the other hand, each path presents its own specific physical phenomena such as free space loss, reflection/transmission, diffraction at large objects, scattering at small objects and waveguiding. [4]

As shown in Figure 2.8, the effects multipath propagation and the channel have on the received power are divided into three categories. The average attenuation, already introduced in subsubsection 2.3.1.2, is generated mostly by propagation loss and presents certain deviations known as fading. These are caused either by large objects between the communicating agents (shadowing/large-scale fading; changes slowly) or dur to interference provoked by the interaction between paths (small-scale fading; very localized). [4]

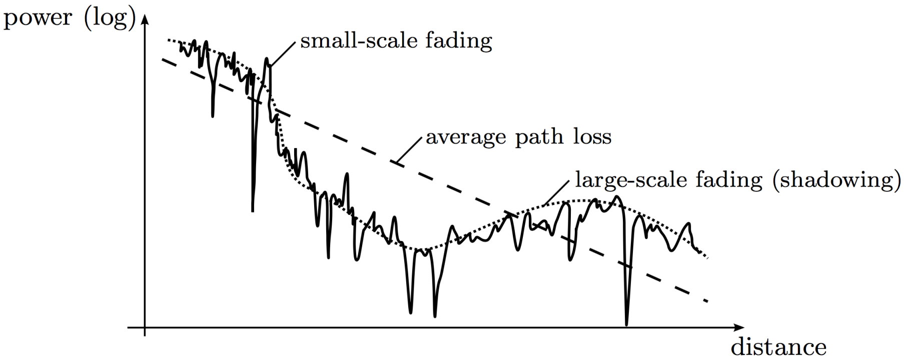


FIGURE 2.8: Channel basics: Comparison of path loss models [4].

In the case of RFID inside an airport, there is little chance we will have large-scale fading and therefore small-scale fading will be the most present effect in the signal apart from path loss. For narrowband systems, there are different channel models for small-scale fadings, such as Rayleigh or Nakagami [72]. However, the most suitable model for UHF RFID in closed spaces with temporal variations due to people has been shown to be the Rician model [73][74].

2.3.2 Data Link Layer

The purpose of the Data Link Layer is to provide the functional and procedural means to transfer data between network entities and to detect and possibly correct errors which may occur in the Physical Layer [66].

In this layer is where the Media Access Control (MAC) protocols are defined. In situations where multiple agents try to communicate with a specific base station, the MAC protocols conform a set of guides and rules aimed to control the communication process for avoiding collisions when possible. These protocols are essential in situations like multiple people making mobile phone calls through the same cellular station or multiple RFID tags trying to reach the same RFID reader.

Specifically, for the case of RFID, the most used protocol is Slotted ALOHA, S-ALOHA. S-ALOHA is a time-based random access protocol in which the RFID reader provides a communication time frame divided in a specific number of identical slots, N. Then the reader broadcasts a message to all tags in range, indicating the number of slots provided, the duration of each slot and requesting the ID of each tag. After receiving the message, the tags respond with their IDs, each tag randomly selecting one of the time slots to communicate.

S-ALOHA is similar to another famous time-based MAC protocol, the TDMA, but unlike in TDMA, the slot assignement is random, whereas in TDMA the base station assigns the slots in a deterministic fashion. This difference is due to the fact that RFID readers *a priori* do not know the amount of tags in range. Then, both protocols make use of the same frequencies for the communication process [75]. As introduced in the previous section, the carrier frequecy lies in the UHF band.

The result of the communication process is a received frame each slot either empty, filled with one tag's ID, or corrupted data caused by two or more tags trying to communicate at the same time (they randomly choose the same slot). Therefore, a reader may need more than one ID request in order to correctly identify all tags in range. In each of these cycles, once a tag is identified it is ignored until the complete set of cycles is finished. When starting a new set, the memory of the reader will be "erased" and all tags will need to fill a slot alone again in order to be identified.

The efficiency of the S-ALOHA strongly depends on the number of slots provided N and the number of tags in range n. As an example we introduce Figure 2.9, where the first frame represents the empty frame the reader provides. The second frame relates to a situation in which a few of the slots and filled with one ID —green— and the rest are received empty. This situation corresponds to the case in which $N >> n$. The third frame relates to a situation where $N \approx n$ and most of the slots are filled with one ID, although we can observe empty slots and collisions —red— too. Finally, in the fourth frame collisions dominate, indicating a situation in which $N << n$ and suggesting an increase in the frame size.

Finally, in the Data Link layer, other operations such as the error control and correction are also carried out. The whole network is simple: Multiple devices trying to communicate with a single node through individual links. This makes us focus our attention on the two layers just presented, as other layers such as the Network Layer or the Transport Layer do not play a significant role in this system. However, they do require attention if we analyze the whole airport network system, where the RFID subsystems become the leaves of a bigger network. We do not deepen on this topic in this Thesis.

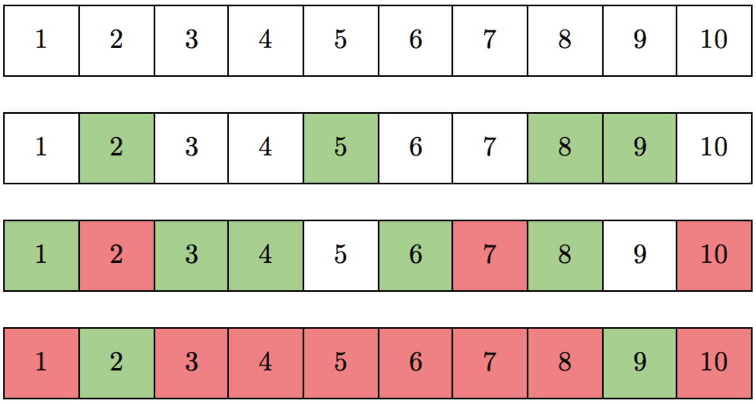


Figure 2.9: Graphical examples of the efficiency of the S-ALOHA protocol.

2.3.3 Regulations

It is well-known every device with the ability to transmit information can not do so freely, as it has to respect some regulations regarding frequency bands or power transmission. And RFID is no exception. Every RFID system has to follow a set of rules or protocols, which vary depending on the geographical area of the planet.

As in several research studies on RFID [76][77], we will focus this Thesis on EPC Class-1 Generation-2 protocols. C1G2 is defined in both the physical and the data link layer, defining standards for several of the RFID communication parts such as the ASK modulation or the S-ALOHA protocol. However, given our system fits in the category of Dense RFID Reader Environment, we are interested in the regulations of the physical layer, as the interference between multiple RFID readers is controlled by protocols in this layer.

C1G2 standard defines RFID communication within the range of 860 MHz to 960MHz, corresponding to UHF. Specifically, given this study is based in North America, we have to consider the standards defined by FCC too. FFC defines the RFID communication to 50 channels of 500 kHz each, all lying in the 902-928 MHz ISM band. FCC also limits the channel occupancy to 400 ms, thus making the RFID readers constantly change their carrier frequency following a mechanism known as *frequency hopping*. In order to avoid frequency interference the readers must ”listen before talk”, by interrogating the channel for 5 ms in order to ensure no other reader is trying to access the same channel. Once a reader has checked a channel is empty, it can begin the tag ID request.

Regarding power restrictions, in North America the maximum transmit power by an RFID reader is 1 W. The transmitting antenna of the RFID reader is limited to a maximum gain of 6 dBi, giving a maximum total radiated power of 4 W EIRP.

2.4 Architectural Decisions

We have reviewed how RFID works and which are the main factors that affect the performance of RFID communication it is time to come up with a set of *interconnected decisions* that capture the essence of the problem and form the different architectures. These decisions are an intermediate system, between the different needs and the final architecture [25].

We will divide the decisions in functional and formal decisions, the former dealing with the functional part of the system, i.e., tracking passengers inside an airport; while the latter focusing on the form of the system, which is the RFID technology *per se*.

2.4.1 Functional Decisions

The functional decisions or decisions in the functional domain are related with form of the solution-specific concept of the system, which is tracking passengers. In section 2.2 we saw that in "tracking passengers", "tracking" is the specific process while "passengers" is the specific operand. We have to work on these in order to develop the functional decisions.

If we look at the process, "tracking", we realize we are in front of a binary process, as either you track someone or not. There is no middle point between tracking someone or not, it is just necessary detecting two times a person in order to have a track of that person. In fact, airports are currently tracking passengers if we follow this definition, as passengers are already being detected or identified at specific points in the terminal. These points or locations are the check-in counters, the security point, the passport control or the gates. So, we have a simple and discrete track of passengers already. The process is completely defined.

Let's focus on the operand then. One thing we have not discussed yet is the fact that there are different types of passengers that travel in different directions inside the terminal. We are talking about the passengers that depart from the airport and the passengers that arrive to the airport. The first question arises: Do we want to track both types of passengers? Furthermore, there is a third type of passengers, the connecting passengers, which we have to decide whether we want to track them or not. We have reached the first set of decisions:

- **Decision 1**: *Do we track departing passengers?*
- **Decision 2**: *Do we track arriving passengers?*

- **Decision 3**: *Do we track connecting passengers?*

All three decisions have just two options, either tracking them or not. Note that deciding tracking a type of passengers increases the information we have on the airport, which will be related to the performance, but also increases the cost, as more tags must be provided.

So far, we know we want can decide to track up to three types of passengers displacing inside the same physical space. If they move through the same space, the difference must be due to how they move in this space. Indeed, all three types of passengers perform different types of general trajectories inside an airport, as the goals are different.

Figure 2.10 shows the general trajectory of a departing passenger, which starts at the parking or drop-off area and ends —hopefully— inside a plane. If the passenger takes an international flight a passport control is required. We can observe that in this process the passenger is identified four times, having a simple tracking system with four data points, not enough for our goal.

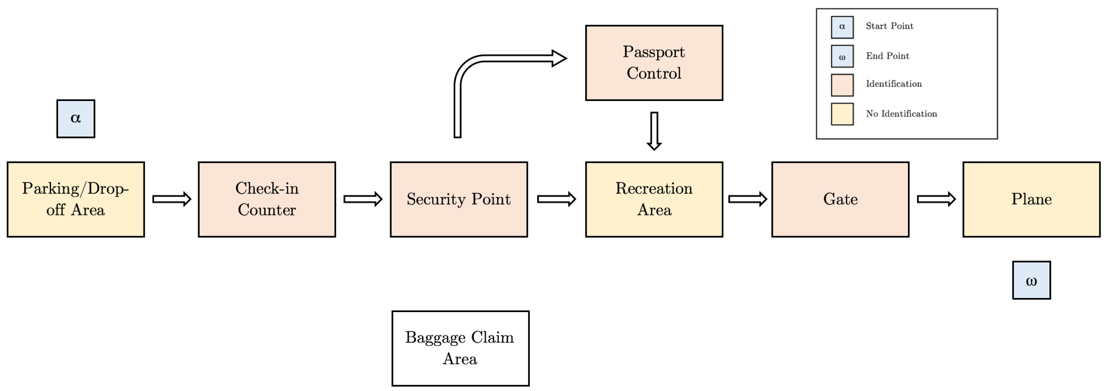


FIGURE 2.10: General trajectory followed by departing passengers inside an airport terminal.

The same diagram for the arriving passengers is shown in Figure 2.11, in which passengers start inside the plane and finish at the parking or drop-off area. In this case, however, the passengers circulate through a new location, the baggage claim area, and are not identified at any point in their displacement.

Finally, Figure 2.12 shows the general trajectory for the connecting passengers, which can be understood as a mixture of the previous two, as passengers start and finish inside a plane. This time the number of identifications is reduced to two if the flight is international or one if it is not, therefore currently the tracking is not ensured for connecting passengers.

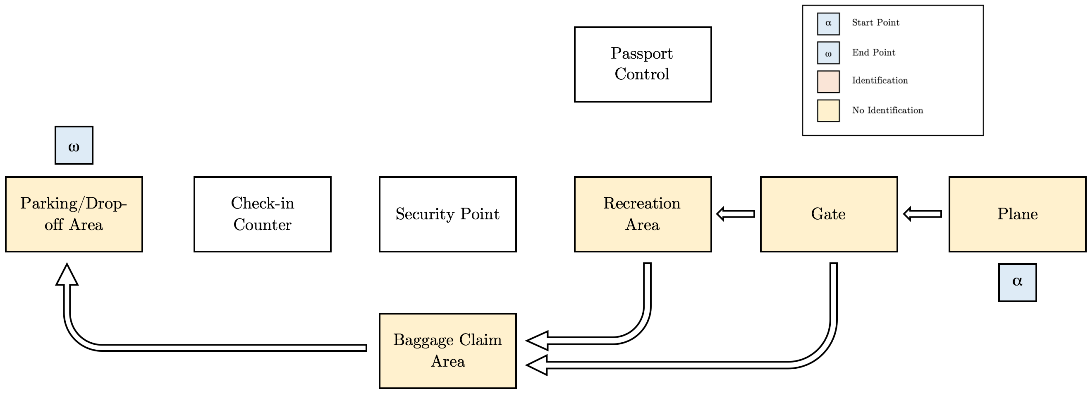


FIGURE 2.11: General trajectory followed by arriving passengers inside an airport terminal.

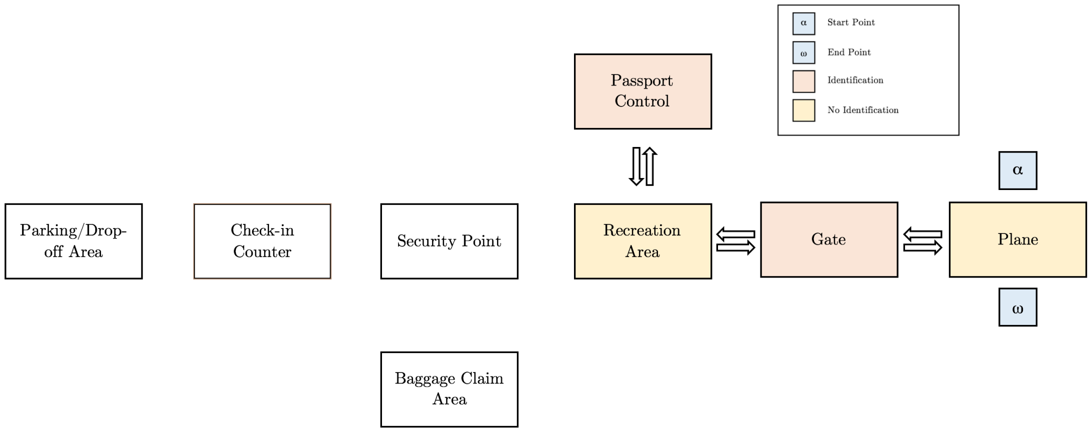


FIGURE 2.12: General trajectory followed by connecting passengers, once on outbound, inside an airport terminal.

After presenting the three general trajectories we realize departing and connecting passengers share the same physical space while arriving passengers have trajectories that pass through different locations. This means that if we want to track both departing and arriving passengers we have to cover more surface with readers than the case of tracking departing and connecting passengers at the same time. If we wanted to avoid spending this extra money in tracking the arriving passengers we could decide to stop the tracking before reaching the baggage claim area. This arises two new questions: Where should the tracking begin? And where should the tracking end?

Again, as the departing and arriving passengers do not share the same space completely, we could design a pair of decisions for each case and consider two tracking directions. As a reminder, the connecting passengers share the physical space completely with the departing passengers, and therefore would not be necessary to specify a pair of decisions for this type of passengers. Therefore, the next set of decisions is:

- **Decision 4**: *Which is the initial tracking point for departing passengers?*

- **Decision 5**: *Which is the final tracking point for departing passengers?*
- **Decision 6**: *Which is the initial tracking point for arriving passengers?*
- **Decision 7**: *Which is the final tracking point for arriving passengers?*

In order to select the options for each decision we have to take a look at the general trajectories for each type of passengers again. Regarding the departing case we have seven different locations the passenger goes through. Out of these seven we discard the recreation area, as it is a very wide area compared to others, and the passport control, given not all passengers go through it. So, we have five potential locations where we could start the tracking and/or finish it. Furthermore, we can realize there is no much sense in tracking passengers from the gates to the planes, as once a passenger enters the gate he/she can not follow any trajectory but the only one possible and has no other choice than waiting at the queue.

So, considering we can not start and finish the tracking at the same location, and therefore the gate can not be the start location for decision 4 and the parking/drop-off area can not be the finishing location for decision 5, the options for these decisions are:

- **Options for Decision 4**: Parking/Drop-off Area, Check-in Counter, Security Point and "None".
- **Options for Decision 5**: Check-in Counter, Security Point, Gate and "None".

Note we introduce "None" for completion and accordance to decision 1, as there is no sense in determining tracking start and end points if we decide not to track in the first place.

Following the same procedure, we can develop the options for decisions 6 and 7, related to the arriving passengers:

- **Options for Decision 6**: Baggage Claim Area, Gate and "None".
- **Options for Decision 7**: Parking/Drop-off Area, Baggage Claim Area and "None".

Considering we have already developed seven decisions, we will focus on the formal domain, as we do not want to come up with too many decisions and increase the variability substantially. In Table 2.5 we summarize the decisions in the functional domain.

Architectural Decision	Option 1	Option 2	Option 3	Option 4
Tracking Departing Passengers (TDP)	Yes	No		
Tracking Arriving Passengers (TAP)	Yes	No		
Tracking Connecting Passengers (TCP)	Yes	No		
Departing Initial Tracking Point (DITP)	Parking/Drop-off Area	Check-in Counter	Security Point	None
Departing Final Tracking Point (DFTP)	Check-in Counter	Security Point	Gate	None
Arriving Initial Tracking Point (AITP)	Baggage Claim Area	Gate	None	
Arriving Final Tracking Point (AFTP	Parking/Drop-off Area	Baggage Claim Area	None	

Table 2.5: Set of functional decisions for the RFID system.

2.4.2 Formal Decisions

The formal decisions or decisions in the domain of form are related with the form of the solution-specific concept, which are an RFID tag and an RFID reader. The performance of an RFID system is notably dependant to the technology used. In [59] is showed how by just changing the tag used, the range of the communication is affected. Our goal in this section is define a set of decisions that remove this technology-dependency as much as possible.

In [76] is showed how the performance is substantially affected by changes in the parameters of the physical layer. Therefore, we will focus the decisions on this layer as much as possible.

In order to define the decisions it is necessary to examine the available technology and check to which extent we are flexible when purchasing parts of an RFID system. Advised by experts on RFID we decide to focus on an RFID supplier's products [78] (see Appendix D).

We start by discussing the RFID readers available (see section D.1). The readers contain the processing unit that handles all the information received by the tags, but do not include the TX and RX antennas, they have be acquired separately. The supplier also offers TX-and-RX-integrated antennas that connect through coaxial cables to the readers (see section D.2). A reader can be connected to up to four antennas, each providing a new area of coverage but increasing the cost. This trade-off can be translated into a new decision:

- **Decision 8**: *How many antennas should we connect to the readers?*

In literature we can find several figures of typical antenna gain diagrams [79][80][81] and the antennas in the supplier's catalog are no different. This type of UHF antennas are usually directional, i.e., they do not cover a perfectly circular area around them but a big lobe area in front of them and a tiny lobe area behind them. In Figure 2.13 the radiation plots of one of the antennas considered are displayed.

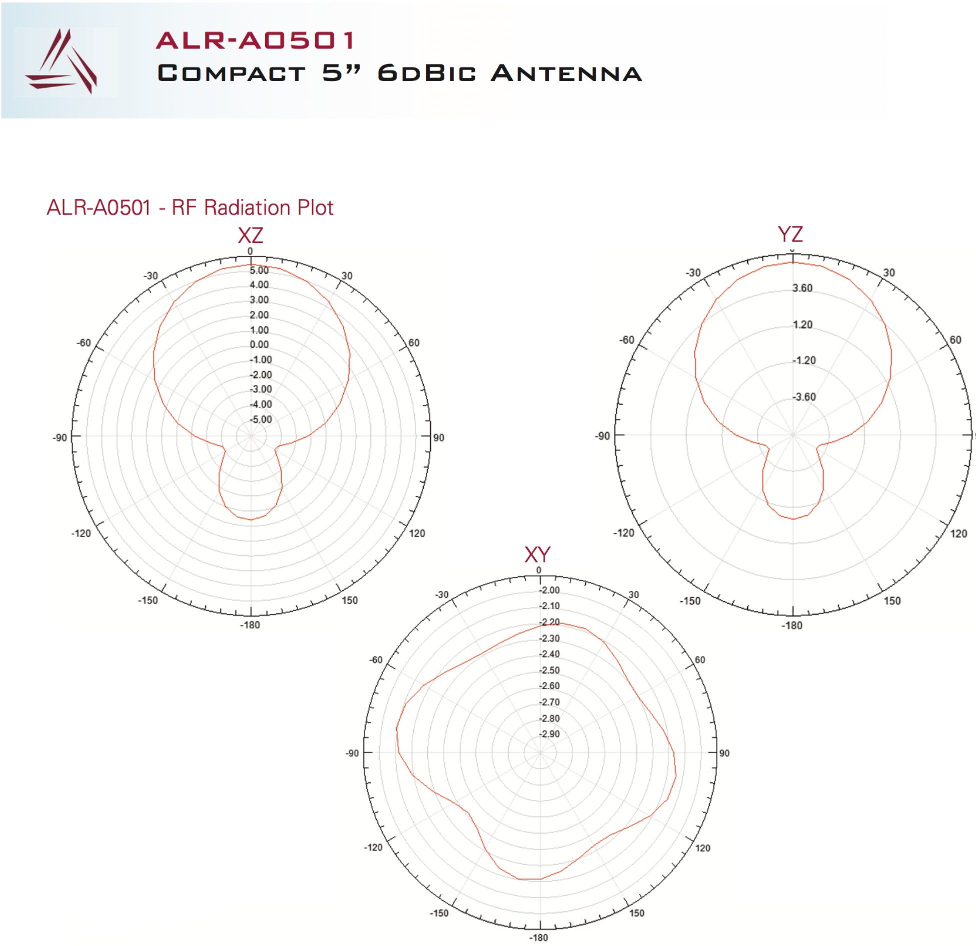


Figure 2.13: RF Radiation Plots fora single ALR-A0501 antenna.

What a radiation plot presents is the gain in every possible direction an antenna radiates. In subsection 2.3.3 the restrictions on the maximum power transmitted allowed were presented. Considering now the radiation plot, we know that the maximum power allowed, which in North America is 4 W, must be transmitted in the direction of the maximum antenna gain. This way, we make sure we do not exceed and fail to fulfill the standards.

In order to achieve omnidirectionality we must connect more than one antenna to the reader, therefore the options for decision 8 turn to be obvious:

- **Options for Decision 8**: 2 antennas, 3 antennas and 4 antennas.

In subsubsection 2.3.1.2 we introduced how the communication in an RFID system was limited by the downlink communication, therefore a key treat of our system is the power sensitivity of the tag. The power sensitivity indicates the minimum power a tag can detect in order to correctly communicate back. Given the antenna do not radiate the same power in all directions, if a tag is located in a direction forming a big angle with the direction of the maximum, it will have more chances of not being reached by the reader.

Therefore, the next step is examining the tag catalog of the supplier, which can be found in section D.3. According to Figure D.3, each tag can be described by two categories: The application it serves and the type of integrated circuit (IC) it has. In this case, we can not choose the application of the tag, as it is a restriction of the system (e.g., we could not select tags designed for pharma/healthcare). On the other hand, one of the things in which the integrated circuits differentiate from each other is precisely the power sensitivity (see section D.4). This introduces us to our next decision, which has three clear options:

- **Decision 9**: *Which IC should the tags contain?*
- **Options for Decision 9**: *Higgs-3*, *Higgs-4* and *Higgs-EC*.

At this point the reader may ask why we do not cover the entire terminal with a great number of readers or just the sufficient number to reach each possible position, so we can forget about omnidirectionalities and integrated circuits. Unfortunately, apart from being a solution that substantially increases the cost, again we are limited by the regulations presented in subsection 2.3.3. In North America, only fifty 500 kHz channels are available for RFID communication, and in [7] we learn there are minimum distances the antennas have to keep in order to avoid interchannel or intrachannel interference.

Although this distances vary with the technology we can have a grasp of its order and why we can not place a lot of antennas, and therefore readers, in the same area. This arises a question which easily turns into a decision:

- **Decision 10**: *Which heuristic should we follow to place the readers?*

Channel Adjacency	Antenna of Default Configuration		
	Front (m)	Side (m)	Back (m)
0	1400	350	210
1	180	45	30
2	130	25	15
3	95	20	10

Table 2.6: Safe distance for different antenna configurations [7].

It is not trivial to develop a good set of options for this question. In the next chapter we will discuss the procedure for obtaining them. For now, for completion, we introduce them as numbered heuristics. So:

- **Options for Decision 10**: Heuristic 1, Heuristic 2, Heuristic 3 and Heuristic 4.

With the three formal decisions presented we have removed a great part of the technology-dependency of our problem. All of the three decisions have their impact on the physical layer of the RFID communication. In subsection 2.3.2 we viewed how the S-ALOHA protocol, lying on the data link layer, had also impact in the communication process, as the guarantor of a correct multiple access.

It was introduced how an ID request from the reader worked and how it usually did not get all tags at once, as the reader might fail in providing the sufficient amount of slots, provoking collisions in some of these slots. This is solved by carrying out more request cycles until the reader considers it has reached all the tags in range with a certain confidence (remember the reader never knows with total certainty the total amount of tags). One may think about providing an very large number of slots in order to avoid collisions. The truth is increasing the number of slots also increases the time elapsed during a request cycle. Imagine a passenger crossing a reader's coverage area while it has just started reading tag IDs from a very large frame. This passenger may get out of the range before the reader can start another cycle, thus missing a passenger. So it is better to perform more than one short cycle rather than just a long big cycle. But, how many cycles should a reader perform before starting another set of cycles?

- **Decision 11**: *How many cycles should a reader perform before starting a new set?*

Ideally, we would want to select a number of cycles that made sure no passenger is missed. Unfortunately, the randomness of the problem makes there is always a probability of missing passengers. In order to come up with an appropriate number of cycles let's study the average human walking behaviour.

We will consider the airport as an environment where most of the passengers walk as if they were walking on the streets, as most passengers arrive to the airport with extra time and do not need to rush. The average adult human walking speed lies around 1.5 m/s [82][83]. Given the examples presented in [59], we would like to aim to a range radius of 10 meters. By doing some easy math we get the average walker would take around 13 seconds to cross a cover area by its diameter.

In [8] the average cycle duration given a fixed number of slots is discussed and we present it in Table 2.7. There, N refers to the number of slots provided, t_N is the duration of a single cycle given a fixed number of slots, σ is the typical deviation in milliseconds for different cycle trials and t_N/t_{N-1} is the ratio between cycle durations of different number of slots.

N slots	1	4	8	16	32	64	128	256
t_N (ms)	56	71	90	128	207	364	676	1304
σ	4.96	2.19	2.26	3.80	4.79	4.82	5.05	4.36
t_N/t_{N-1}	-	1.3	1.3	1.4	1.6	1.8	1.9	1.9

Table 2.7: Cycle duration given a fixed number of slots [8].

This measurements are closely related to the technology used, but again, we can get a grasp about the order of the duration of the cycles. It is fair to assume we will not have as many people as 256 in an approximately ten meter radius circular area, specially considering large flows of people occasionally reach 2 people per square meter [84]. So, it is highly likely that all cycles last less than a second.

If we set the duration limit of a complete set of request cycles to five seconds, we will have enough time to —hopefully— detect a passenger twice while he/she is in range. Assuming a second out of those five is spent performing extra operations (information processing, cycle preparation...), a reader could fit up to four cycles in that time for the worst-case scenario. So, given that and the fact that one cycle is not usually enough —considering the reader provides a reasonable number of slots—, the options we will analyze are:

- **Options for decision 11**: 2 cycles, 3 cycles and 4 cycles.

This last decisions closes the set of formal decisions we will consider for our system. They are summarized in Table 2.8.

Architectural Decision	Option 1	Option 2	Option 3	Option 4
Antennas per Reader (APR)	2	3	4	
Tag Technology (TT)	*Higgs-3*	*Higgs-4*	*Higgs-EC*	
Reader Placement Heuristic (RPH)	Heuristic 1	Heuristic 2	Heuristic 3	Heuristic 4
Number of ALOHA Cycles (NOAC)	2	3	4	

Table 2.8: Set of formal decisions for the RFID system.

2.4.3 Morphological Matrix

The eleven architectural decisions we have come up with are the subset of design decisions that are most impactful. We organize them in a structure called morphological matrix, as showed in Table 2.9. The combination of these decisions will lead to different architectures that are fundamentally different from each other.

Architectural Decision	Option 1	Option 2	Option 3	Option 4
Tracking Departing Passengers (TDP)	Yes	No		
Tracking Arriving Passengers (TAP)	Yes	No		
Tracking Connecting Passengers (TCP)	Yes	No		
Departing Initial Tracking Point (DITP)	Parking/Drop-off Area	Check-in Counter	Security Point	None
Departing Final Tracking Point (DFTP)	Check-in Counter	Security Point	Gate	None
Arriving Initial Tracking Point (AITP)	Baggage Claim Area	Gate	None	
Arriving Final Tracking Point (AFTP	Parking/Drop-off Area	Baggage Claim Area	None	
Antennas per Reader (APR)	2	3	4	
Tag Technology (TT)	*Higgs-3*	*Higgs-4*	*Higgs-EC*	
Reader Placement Heuristic (RPH)	Heuristic 1	Heuristic 2	Heuristic 3	Heuristic 4
Number of ALOHA Cycles (NOAC)	2	3	4	

Table 2.9: Morphological Matrix containing the decisions chosen for the RFID system.

One could think that combinatorics tells us there will be 124,416 different architectures to compare. Although the math is correct, in reality this number is not that big, given there exists a set of constraints or restrictions that make some of these architectures unfeasible to carry out. As an example, if we decide not to track departing passengers, there is no logic in installing readers at the check-in counters, therefore making unfeasible all architectures with option 2 for decision 1 and option 2 for decision 4. All these constraints can be found summarized in Table 2.10.

Id	**Name**	**Scope**	**Equation**
a	Tracking	TDP, TAP	(TDP != No) || (TAP != No)
b	TDPconstraint1	TDP, DITP, DFTP	(TDP == No && DITP == None && DFTP == None) || (TDP == Yes)
c	TDPconstraint2	TDP, DITP, DFTP	(TDP == Yes && DITP != None && DFTP != None) || (TDP == No)
d	TAPconstraint1	TAP, AITP, AFTP	(TAP == No && AITP == None && AFTP == None) || (TAP == Yes)
e	TAPconstraint2	TAP, AITP, AFTP	(TAP == Yes && AITP != None && AFTP != None) || (TAP == No)
f	DTPconstraint1	TDP, DITP, DFTP	(DITP != DFTP) || (TDP == No)
g	DTPconstraint2	DITP, DFTP	(DITP == Security Point && DFTP == Gate) || (DITP != Security Point)
h	ATPconstraint1	TAP, AITP, AFTP	(AITP != AFTP) || (TAP == No)
i	ATPconstraint2	AITP, AFTP	(AITP == Baggage Claim Area && AFTP == Parking/Drop-off Area) || (AITP != Baggage Claim Area)
j	DFTP+AITP	DFTP, AITP	(DFTP == Gate && AITP == Gate) || (DFTP != Gate)

TABLE 2.10: Constraints for the RFID system.

With the morphological matrix defined, we have reached the end of this chapter. Our next goal is to develop a simulation model in order to evaluate all feasible architectures and compare them in performance and cost. Also, system architecting is an iterative process, meaning we will need to constantly review our set of decisions and constraints developed during this chapter and, in some cases, redefine them. To ease the comprehension of the refinement process, Table 2.11 shows each architectural decision and the section of the following chapter in which it will be addressed and, if necessary, refined.

Architectural Decision	**Section**
Tracking Departing Passengers (TDP)	subsection 3.2.2
Tracking Arriving Passengers (TAP)	subsection 3.2.2
Tracking Connecting Passengers (TCP)	subsection 3.2.2
Departing Initial Tracking Point (DITP)	subsection 3.2.1
Departing Final Tracking Point (DFTP)	subsection 3.2.1
Arriving Initial Tracking Point (AITP)	subsection 3.2.1
Arriving Final Tracking Point (AFTP	subsection 3.2.1
Antennas per Reader (APR)	subsection 3.3.2
Tag Technology (TT)	subsection 3.4.1
Reader Placement Heuristic (RPH)	subsection 3.3.3
Number of ALOHA Cycles (NOAC)	subsection 3.3.4

Table 2.11: Relation of each architectural decision with the section in which it will be addressed.

Chapter 3

Model Description

After defining the different architectural decisions and determining which is the set of feasible architectures based on restrictions imposed by the problem, it is time to evaluate each of these architectures. Regarding the Four Main Tasks of Decision Support Systems in [25], the goal of this chapter is to focus on the *simulating* layer and present the simulation program developed to evaluate the different architectures.

The chapter is organized as to introduce the different parts of the code in a structured way. To that end, section 3.2 is devoted to the airport environment, i.e., how the program models the physical space of the terminal and the different passengers walking through it. In section 3.3, how to simulate the communication process between readers and tags is discussed, focusing on the impact each decision has in the program. Finally, in section 3.4 the metrics used to evaluate the different architectures will be presented.

3.1 Diagram of the Model

To help understanding how the model works and how the different parts that will be presented in this chapter relate to each other, in Figure 3.1 a diagram of it is presented.

In blue, the different inputs of the whole model are showed, being the flight schedule (presented in section 4.1), the airport map and the architectural decisions. This chapter will introduce first the Airport and Passenger models in subsection 3.2.1 and subsection 3.2.2, respectively. Then, in section 3.3, the Coverage, Reader, Tag and ALOHA models will be explained in detail. Finally, the result of the simulations or the architectures' metrics, will be introduced in section 3.4.

Note that the Passenger and Airport models only are called once during the entire simulation, as all the architectures share the results of both models, as they represent

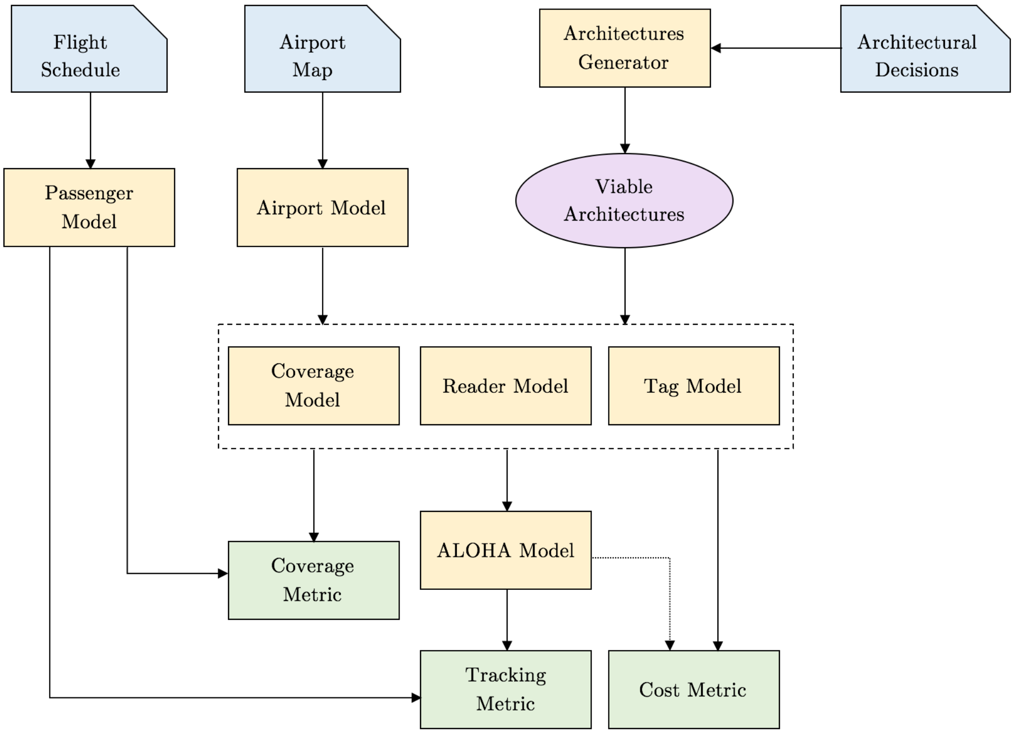


FIGURE 3.1: Diagram of the model showing how the different parts interact with each other to obtain the final metrics.

the context of the system. Instead, all the models dealing with the RFID part are called exactly once per architecture, and form the basis for the performance and cost metrics.

3.2 Environment

The first essential step of the model is the creation of the system's context, which is the airport terminal itself. This means simulating not only the physical space of the terminal with a reasonable level of detail but also the movement of the passengers through it.

Terminal simulators have already been developed in literature with the aim of providing insightful information for facility design [85] or optimizing passenger flow [86]. In our case, we want to simulate the airport and then simulate an RFID communication system on top of it, thus differing from the usual goals of airport simulators.

Note this is the part of the model that all architectures share, as no decision involves changing any feature of the airport. Therefore, the environment modelling will not be as critical as the RFID modelling, which will vary between the different architectures. However, it is important to model the terminal and the passengers appropriately in order the results of the RFID simulation to be valuable.

3.2.1 Terminal

The airport constitutes the physical space, i.e., the form, of the context of our system and the task of this section is to model this physical space with the necessary level of detail. To begin with, we will focus just on the airport passenger buildings, as neither the plane runways nor other locations such as the car rental area will be populated with RFID readers. Considering our exclusive focus on the building and in order to generalize the architectures as much as possible, we redefine the fourth decision, regarding to the Departing Initial Tracking Point, DITP. Instead of considering the parking or drop-off area as an option we will change it for just the entrances of the building. This change is reflected in Table 3.1.

DITP	**Option 1**	**Option 2**	**Option 3**	**Option 4**
Redefinition	Entrance	Check-in Counter	Security Point	None

TABLE 3.1: Redefinition of the Decision *Departing Initial Tracking Point.*

The passenger building is a 3D environment, but the passengers walk across 2D planes in most cases. In few situations such as in elevators —closed environment— or on mechanical stairs the passengers vary their altitude. As this situations suppose a small part of the passengers' trajectories we will model the airport as a flat surface, following the same concept of other airport simulators [87].

As stated in [6], "Designers of airport passenger buildings face a fundamental problem: They need both to concentrate and spread them out". Given designing an airport is not an easy task we opt for focusing on the five basic airport passenger building configurations, also from [6]. These configurations are:

- Finger piers
- Satellites, with or without finger piers
- Midfield, either linear or X-shaped
- Linear, with only one side devoted to aircraft
- Transporters

In Appendix E each of the configurations is described with further detail. The first two configurations present a major level of complexity whereas the others are much simpler. Motivated by this fact, and the difficulty of designing the interiors of the buildings from scratch, we decide to model a real airport, in 2D, whose configuration lies in either the Finger pier or Satellites categories.

Based on the previous reasoning, and motivated by the amount of data available from the airport, we choose to model the Terminal 3 at Chicago O'Hare International Airport (ORD). O'Hare airport is one of the busiest airports in the world, as it serves as a hub connecting several destinations between the east and the west of America. The airport serves an average of 2,400 flights per day and it is divided in 4 terminals and 9 concourses [88].

Four of the nine concourses are located in Terminal 3, which hosts domestic flights from several well-known airlines. Observing Figure 3.2 we can appreciate the clear resemblance of the terminal with the Finger Pier model, as the building shows long, narrow walkways extending from the central core and aircraft gates spread across them. Each of the fingers corresponds to a concourse of the terminal. From left to right, G, H, K and L, respectively.

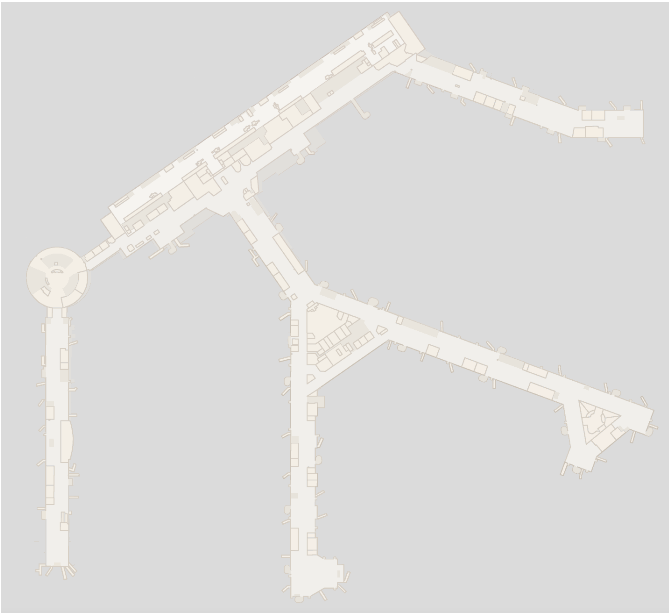


FIGURE 3.2: Real map of the Terminal 3 at Chicago O'Hare International Airport (level 2) [5].

Furthermore, the terminal has three different levels: In level B1 we can find the parking area, level 1 hosts the baggage claim area and in level 2 all gates and check-in counters are located. Levels B1 and 1 do not have fingers and passengers can only walk through the central core, thus having less floor surface. Given the baggage claim area is located in a relatively small space, with no presence of cafés, restaurants and other businesses such as those that fill level 1, it narrows the space of trajectories to those that arrive at level 1 and go to the exit of the terminal, with possible stops at the conveyor belt and/or the restrooms. This difference in the richness and variety of trajectories between different levels makes us focus solely on level 2.

This change makes us reconsider the morphological matrix again, as now the option of installing RFID at the baggage claim area has been discarded. So, we have to redefine the decisions regarding the arriving passenger's initial and final tracking points. The redefinition —or specialization— of these decisions is showed in Table 3.2, where now we do not consider placing readers between the baggage claim area and the exit.

Architectural Decision	**Option 1**	**Option 2**	**Option 3**	**Option 4**
Arriving Initial Tracking Point (AITP)	Gate	None		
Arriving Final Tracking Point (AFTP	Baggage Claim Area	None		

Table 3.2: Redefinition of AITP and AFTP decisions.

Note now this pair of decisions is redundant with the *tracking arriving passengers* decision, as it also can be viewed as a binary yes/no decision. Therefore, we remove the AITP and AFTP decisions from the morphological matrix.

To model the layout of level 2 as a flat surface we view the map as a discrete domain composed by tiles, and assign each of these tiles a different number depending on its nature. So, we define the terminal T as

$$T = (t_{ij}) \in \mathbb{R}^{m \times n} \tag{3.1}$$

The chosen values for m and n will be discussed in the next section. The different tiles are labeled depending on whether they are walking areas, obstacles or the outside of the terminal. Therefore, specifically, we will consider that

$$t_{ij} \in \left\{0, \frac{1}{2}, 1\right\} \tag{3.2}$$

where 0, 1/2 and 1 correspond to the outside area, the obstacles and the walking area, respectively. In addition, the different locations l_k —forming a set L— of the terminal T that are part of passengers trajectories, i.e., entrances, check-in counters, security points, gates and exits, are modeled as a tuple

$$l_k = (i_k\,,\, j_k\,,\, c_k) \tag{3.3}$$

where i_k and j_k are the coordinates of the location's tile in T and c is its class, i.e., the type of location among the listed in the previous paragraph. A location occupies exactly one tile. In Figure 3.3 the terminal 3 of O'Hare airport is presented following the modeling described in this section. Regarding the terminal T, black areas, with $t_{ij} = 0$, correspond to the outside of the terminal; white areas, in which $t_{ij} = 1$, represent the walking areas; and finally gray areas, with $t_{ij} = 1/2$, model the obstacles in the terminal. These obstacles may be closed spaces, such as shops or restaurants; or restricted walking areas, such as the security control areas.

FIGURE 3.3: Model of the Terminal 3 at Chicago O'Hare International Airport.

On the other hand, colored dots in Figure 3.3 represent the position of each of the locations l_k. Thirteen Red dots represent the different entrances, check-in counters are showed as twenty-eight blue dots, ten green dots relate to where the various security control points are located, seventy-eight yellow dots represent the gates of the terminal and finally a single purple dot models the exit of the terminal for the arriving passengers, which in reality would lead them to level 1. As Terminal 3 only serves domestic flights we do not include a passport control area.

Note the fingers of the terminal are displayed as horizontal lines instead of diagonals, the true design of the airport. Again, given no airport feature is involved in the morphological matrix, we opt for simplifying the design in order to ease the creation of the simulation model. However, we do not change the relative positioning of obstacles within the fingers, although we cluster them in bigger gray areas also for simplicity. The terminal and the remaining parts of the model to be presented in the following sections are developed using Python coding language.

3.2.2 Passengers

In this section we introduce how we model the passengers that depart from or arrive to terminal T, the behaviour of their trajectories and their inclusion within a time frame.

3.2.2.1 Terminal Granularity

We first focus on the physical space of the terminal and how the passengers move through it. In order to start discussing how to model the different trajectories inside the airport, it is essential to determine the granularity of the terminal T, i.e., provide numerical values to the variables m and n introduced in the previous section.

There is a tradeoff between the size of the tiles and the amount of computational resources required to carry out the simulation. If we chose a small tile size —and therefore large values for m and n— the simulation would be computationally expensive whereas a large tile size would ease the computing but would not provide an accurate level of detail to the terminal.

In subsection 2.4.2 we stated our objective of achieving a range of 10 meters approximately and we have to set tiles' size accordingly to this. Therefore, a size larger than 1 meter would not be suitable for the case, as we would not be providing the necessary level of detail. On the other hand, setting 10 centimeters as the size of each tile would not only provide too much detail but also increase the computing substantially.

Given we have to model passengers walking, a good way to make all the calculations simple is to make the tile size equal the human average stride length. This way, a passenger moving from one tile to an adjacent one would be interpreted as this passenger actually making a step. From literature, we learn the average stride length is 0.76 meters [89]. So, taking into account the real dimensions of O'Hare's Terminal 3 and adding some margins for aesthetics we choose

$$m = 600 \qquad n = 725$$

Therefore, we end having a terminal of 435,000 tiles which represents the entire terminal surface.

3.2.2.2 Passenger Trajectories

Inspired by the work in [90], we model trajectories as a succession of different locations. For instance, departing passengers first get inside the terminal, may head to the check-in counter, then walk towards the security point and finally reach the corresponding gate. They may deviate during their trajectories but they will not visit a specific location l_k before reaching all previous locations showed in Figure 2.10.

Formally, a passenger's l trajectory PT_l is defined as set of steps s_{lk}

$$PT_l = (s_{l1}, s_{l2}, \cdots, s_{lK}) \tag{3.4}$$

Where each step s_{lk} is defined by its coordinates

$$s_{lk} = (i_{lk}\, , \, j_{lk}) \tag{3.5}$$

While reaching each of the locations, passengers are only able to walk through walk areas, and can not cross obstacles or move outside of the terminal. It is also obvious that if a passenger's step s_{lk} has specific coordinates (i, j), his/her next step s_{lk+1} must be either $(i + 1, j), (i - 1, j), (i, j + 1), (i, j - 1), (i + 1, j + 1), (i + 1, j - 1), (i - 1, j + 1)$ or $(i - 1, j - 1)$. This means no passenger can move to any tile that is not adjacent to his/her current tile.

According to the previous definitions, the developed algorithm for computing a passenger's trajectory starts at the first location the passenger has to visit and makes a succession of steps to the next location or target. Once the passenger reaches this next location, target changes to the following location until the passenger arrives to the final location, being the gate for the departing passengers or the exit for the arriving passengers.

Due to the absence of real data regarding connecting passengers in Chicago O'Hare Airport we decide to focus only on departing and arriving passengers, and therefore will base all trajectories either on Figure 2.10 or Figure 2.11. This means that, for now, we will set the *Tracking Connecting Passengers* decision to "No" for all architectures.

The modeling of human walking has already been studied for numerous purposes and applications [91][92][93]. However, current walking models present such a high level of detail and complexity that using these models to create more than 100,000 people walking distances over 300 meters would take an excessive amount of time and computing resources. Given we do not possess such capabilities we create a simpler long-distance walking model able to handle the scalability issues.

We present algorithm 1 as our approach to computing a human-walking trajectory in the terminal context.

Algorithm 1 compute_trajectory

```
procedure ComputeTrajectory
    next_location ← 1
    step ← 0
    target ← locations[next_location]
    while not at_final () do
        make_next_step (target)
        step ← step + 1
        if position == target then
            if next_location + 1 < length (locations) then
                next_location ← next_location + 1
                target ← locations[next_location]
        if step > 2 · (terminal_lenght + terminal_width) then
            break ()
```

As explained in the previous paragraphs, each passenger has a set of assigned locations, which is represented by the *locations* array. For each type of location, the passenger is assigned one instance randomly. At the beginning of the trajectory, the passenger is placed at the first location in *locations* and sets his/her target on the second location of the array. While the target is not reached, the passenger will keep walking. Once the he/she arrives to the target, the algorithm will provide the passenger a new target until the final location is reached. For instance, the typical *locations* array for a departing passenger looks like

$$locations = [l_{k_1}, l_{k_2}, l_{k_3}, l_{k_4}] = [(i_{k_1}, j_{k_1}, c_{k_1}), (i_{k_2}, j_{k_2}, c_{k_2}), (i_{k_3}, j_{k_3}, c_{k_3}), (i_{k_4}, j_{k_4}, c_{k_4})]$$

Where $c_{k_1}, c_{k_2}, c_{k_3}$ and c_{k_4} are an entrance, a check-in counter, a security point and a gate of the terminal, respectively. Finally, the algorithm has a *break* statement in order to prevent the simulation to get stuck in the creation of a single trajectory. In other words, if the number of steps walked by a passenger is greater than a threshold, the algorithm does not converge to a valid path. The algorithm that decides and makes the next step of a trajectory is presented as algorithm 2.

Basically, we set the passenger to move in the direction that will get him/her closer to the target. However, we know passengers deviate to other places (shops, restaurants, restrooms...) and we model this as a random variable that makes passengers take non-optimal directions. The direction of these latter type of movement can be totally random or present some sort of logic within the randomness. These three types of movement respond to each of the “make-a-step” functions in the algorithm:

Algorithm 2 Decide and Make Next Step

```
procedure MAKE_NEXT_STEP(target)
    if rand() > 0.92 then
        if rand() > 0.005 then
            direction ← get_logical_step (position, target)
        else
            direction ← get_random_step (position, target)
        move (direction)
        for i ∈ {0, ··· , ⌊0.03 · terminal_length⌋} do
            if check_walkway (direction, position) then
                move (direction)
    else
        move (get_closest_step (target))
```

- *get_closest_step*: This function returns the direction the passenger has to follow in order to get the closest to the target. We base the definition of the function on Graph Theory, by reinterpreting the terminal as a connected graph, in which each tile turns into a node and edges are created between adjacent tiles. We then apply the *A** search algorithm [94] in order to find the shortest route from the passenger's current position to the target. We choose *A** algorithm with euclidean distance as guiding heuristic [95]. We make use of Python's `NetworkX` library to get the *A** path.

- *get_logical_step*: This function returns a random direction where no obstacle is present. However, if the passenger is positioned next to an obstacle or wall, the algorithm will substantially increase the probability of returning a direction that makes the passenger head away from that obstacle or wall, as people in airports do not tend to walk next to walls.

- *get_random_step*: This function randomly selects a direction within the set of clear directions, i.e., directions with no obstacles in the way.

Even though we use efficient algorithms, as we have selected relatively large values for terminal sizes m and n, the creation of each trajectory takes more time than expected, causing a time delay we can not afford. To solve this, we propose downsizing the terminal to create all trajectories and then enlarging the terminal and interpolate all trajectories to fit the new size. Being able to enlarge the terminal and the trajectories after creating them is a powerful tool that lets us increase the granularity of the terminal without increasing the computational cost. However, the memory space used does increase.

We take advantage of this new feature and set a final terminal sizes m_f and n_f of 1200 and 1450 —doubling the initial m and n values—, respectively. This way, we will be able to be more precise when determining coverage radii in the coming sections. To compute trajectories we will downsize these values to m_i and n_i, being 120 and 145, respectively, and thus reducing the matrix dimensions by a factor of ten. Figure 3.4 and Figure 3.5 show examples of trajectories that have been interpolated to the resized terminal without losing any detail.

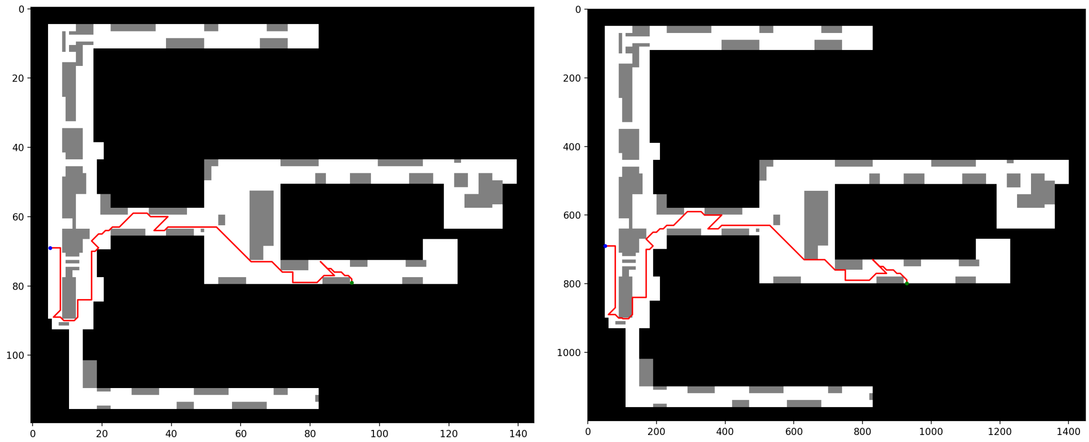


FIGURE 3.4: Departing passenger's trajectory before (left) and after (right) resizing and interpolation.

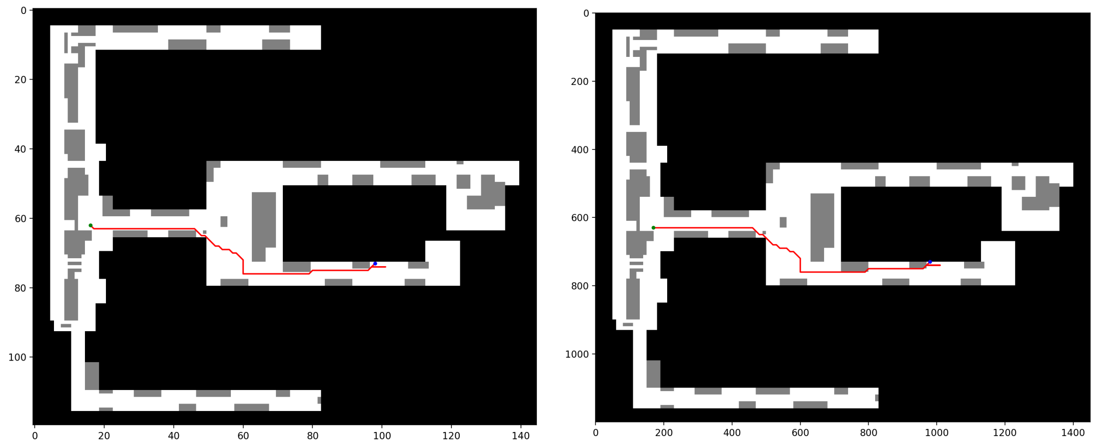


FIGURE 3.5: Arriving passenger's trajectory before (left) and after (right) resizing and interpolation.

Note how trajectories are not perfectly optimal and present notable deviations, specially in the case of departing passengers, that reflect the non-optimal behaviour of human walking trajectories.

3.2.2.3 Temporal Modeling

Apart from creating the passengers in the spatial domain as we have done in the previous section, it is necessary to place them in the temporal domain. It is important to specify when each passenger gets in and out of the terminal taking into account how it happens in a real terminal. For instance, making passengers stay in the simulated terminal T less than they actually stay in the real terminal would lead to a low passenger density at the airport.

To model the time passengers stay inside the terminal we have to consider departing and arriving passengers separately. On the one hand, departing passengers usually arrive more than an hour before the flight and have to spend time waiting for boarding. On the other hand, arriving passengers get inside the terminal the moment they walk through the boarding gate and do not stop until they reach the baggage claim area or the exit of the terminal.

Unfortunately, we lack real data on the temporal behaviour of passengers and can not model this feature as accurately as we would like to. However, we can base our approach on the work already done in [6]. In Figure 3.6, the usual aspect of the cumulative passenger arrival is shown, it represents the total number of arrivals of departing passengers over a period of time.

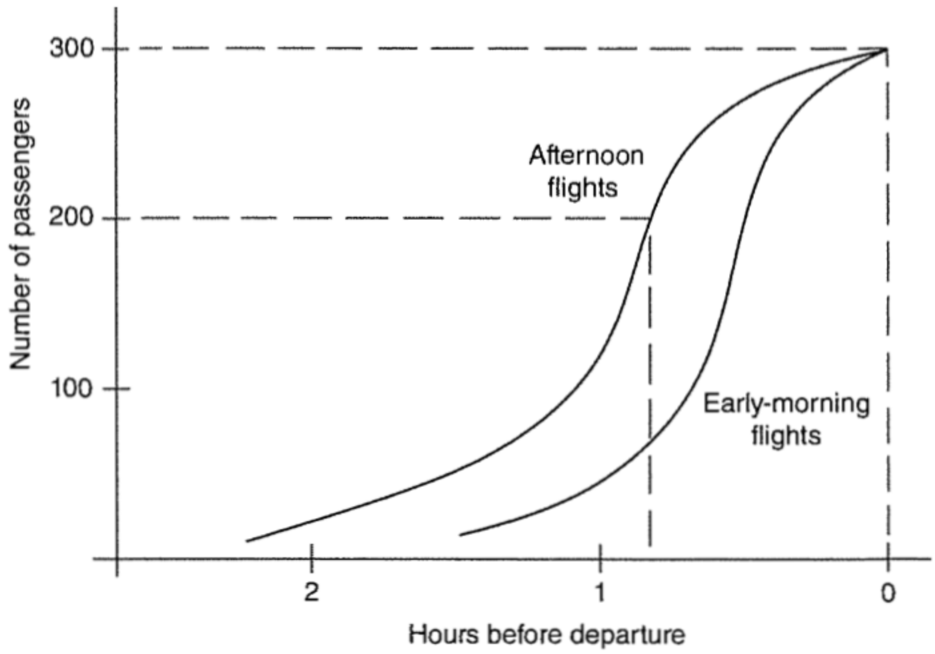


FIGURE 3.6: Arrival diagrams for early-morning and afternoon departures [6].

We develop two heuristics, one for each travel direction —departure and arrival—, by which we try to model how different passengers from the same flight have different arrival times. For the departing case, we define the time spent inside the terminal t_{t_d} as

$$t_{t_d} = walking\ time + t_{early} + t_{b_d} \tag{3.6}$$

where t_{early} relates to the time a passenger has to wait because of the early arrival at the airport and is defined as $t_{early} \sim N(2500\,\text{s},\, 500\,\text{s})$; and t_{b_d} represents the time a passenger spends waiting from the moment the boarding starts until this passenger boards, modeled as $t_{b_d} \sim N(0,\, 240\,\text{s})$.

In Figure 3.7, we show the result of applying both our developed temporal and spatial models in the cumulative passenger arrival time curve for a departure flight of 300 passengers. A passenger is taken into account as soon as he/she enters the terminal and then it is no longer considered when he/she walks through the boarding gate. According to the figure, the first passengers arrive around 75 minutes before the boarding starts, meaning they arrive 2 hours before the flight approximately. Comparing this curve to the curves showed in Figure 3.6 we can appreciate their similarity.

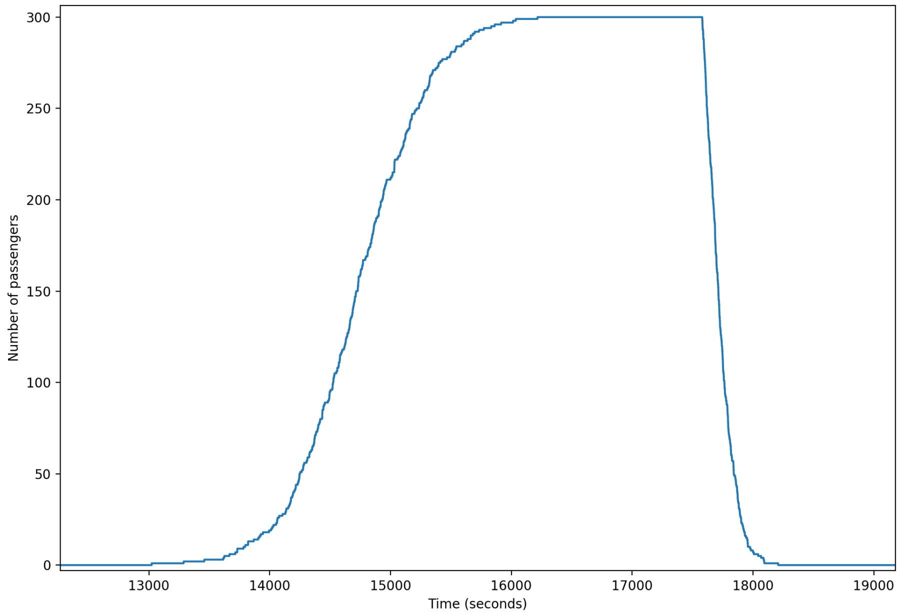


FIGURE 3.7: Cumulative passenger arrival time curve for the departing case.

In contrast, the heuristic developed for the arriving case is much simpler. The time spend inside the terminal for the arriving passengers t_{t_a} is defined as

$$t_{t_a} = walking\ time + t_{b_a} \tag{3.7}$$

Where t_{b_a} represents the different times passengers get out of the plane and therefore walk through the boarding gates. It is defined as $t_{b_a} \sim N(0,\, 200\,\text{s})$, given arriving passengers spend less time queuing than departing ones, as no ticket checking is needed.

Figure 3.8 shows the cumulative passenger arrival time curve for the arriving case. Immediately, we can appreciate the significant difference with the departing case, as usually these passengers walk straight to the baggage claim area without making any stops or having to wait in determined spaces. In the figure it is observed that around 15 minutes pass between the first passenger walks through the boarding gate until the last passenger reaches the baggage claim area. This value strongly depends on the distance between those places.

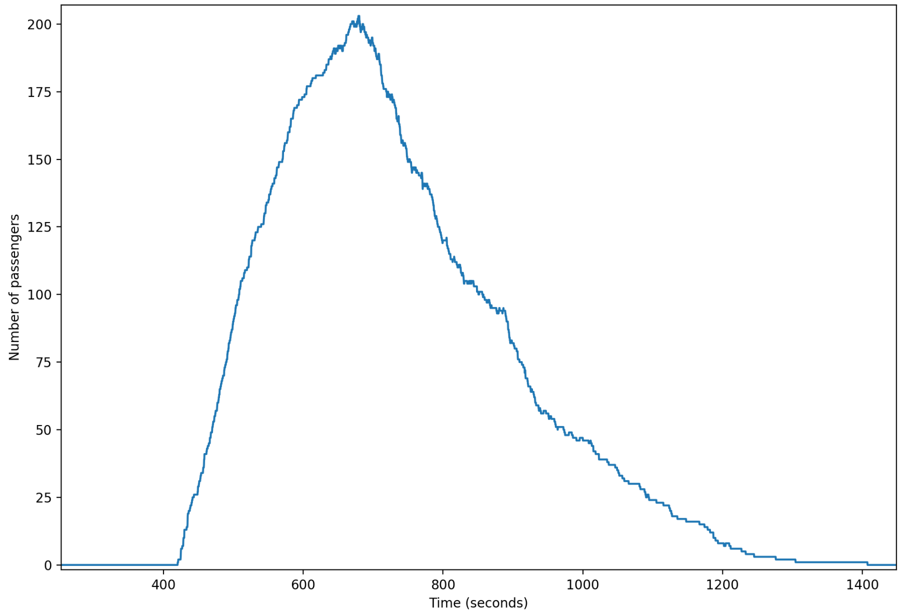


FIGURE 3.8: Cumulative passenger arrival time curve for the arriving case.

Regarding the temporal modeling of the whole simulation, in subsection 2.4.2 we introduced the average human walks at an average speed of 1.5 m/s. As we have set a tile size of half the human stride length, doing simple math we get that the passengers created by the model have to walk four tiles every second. To ease the creation of the model, we manage the simulation's internal clock in a fixed-increment time advance fashion. So, for instance, if we want to recreate an hour of activity, the model will perform 14,400 clock advances.

At this point we conclude the description of the modeling of the terminal environment, the infrastructure all architectures share. This way we make sure that all architecture simulations are based on the same flight schedule, number of passengers and walking trajectories.

3.3 RFID Communications

Once we have discussed the modeling of the terminal and its passengers it is time to decide how we will model each of the different architectures. The key task in this section is determining which implications each of the decisions has on the RFID modeling, which will be based on the RFID introduction presented in section 2.3.

In literature we find studies that have involved the creation of RFID simulators to obtain insightful results. Two powerful examples are, RFIDSim [80], a physical and logical layer engine for passive RFID, and the PARIS Simulation Framework [4], an RFID simulation environment that focuses on the physical layer, providing a detailed model of the UHF communications. Unfortunately, both models are not easy to scale to meet the airport environment requirements without increasing the amount of computational resources needed. Therefore, we create a new RFID model focusing on a scaled version of current simulators in exchange of level of detail.

At the beginning of this Thesis we claimed how System Architecture was an iterative process, as each stage of the project requires a reconsideration and validation of the different architectures. In the previous section we have already redefined or removed some of the decisions because of this iterative behaviour. In Table 3.3 we show the current definition of the morphological matrix.

Architectural Decision	**Option 1**	**Option 2**	**Option 3**	**Option 4**
Tracking Departing Passengers (TDP)	Yes	No		
Tracking Arriving Passengers (TAP)	Yes	No		
Tracking Connecting Passengers (TCP)	No			
Departing Initial Tracking Point (DITP)	Entrance	Check-in Counter	Security Point	None
Departing Final Tracking Point (DFTP)	Check-in Counter	Security Point	Gate	None
Antennas per Reader (APR)	2	3	4	
Tag Technology (TT)	*Higgs-3*	*Higgs-4*	*Higgs-EC*	
Reader Placement Heuristic (RPH)	Heuristic 1	Heuristic 2	Heuristic 3	Heuristic 4
Number of ALOHA Cycles (NOAC)	2	3	4	

TABLE 3.3: Refined Morphological Matrix containing the decisions chosen for the RFID system.

This section will focus on the formal decisions, which are the ones that strongly relate with the implementation of the RFID technology. We divide the section in four subsections, each relating to one of the formal decisions of the matrix.

3.3.1 Link Budget

We proceed in the same way we did in section 2.4 and start focusing on the physical layer of the RFID communication system. One of the unknowns we still have to solve is the effective communication distance between readers and tags. As introduced in previous sections, the range for far-field RFID communication is between 5 and 20 meters, while most readers achieved an average of 10 meters.

One of the tools telecommunication engineers use for estimating the maximum distance of a communication link is the link budget, an accounting of the different gains and losses a signal undergoes during its way from the TX to the RX. So, the power received at the RX side is

$$Power\, received\,(\text{dBW}) = Power\, transmitted\,(\text{dBW}) + Gains\,(\text{dBW}) - Losses\,(\text{dBW}) \quad (3.8)$$

The RX will correctly be able to detect and process the signal if the power received is greater than its power sensitivity. In subsubsection 2.3.1.2 we viewed the critical link for the whole communication process was the downlink, i.e., the situation in which the reader is the TX and the tag the RX. So, if we redefine Equation 2.2 differentiating gains and losses we have

$$P_{rt}\,(\text{dBW}) = P_{tr}\,(\text{dBW}) + G_{tr}\,(\text{dBW}) + g_{rt}\,(\text{dBW}) - PL\,(\text{dBW}) - \chi\,(\text{dBW}) - \tau\,(\text{dBW}) \quad (3.9)$$

We have to use Equation 3.9 to compute the required maximum distance so the power level received by the tag is sufficient. To that end, we make the following considerations or assumptions:

- To compute the maximum distance, the received power P_{rt} has to equal the power sensitivity of the tag, $P_{rt} = S_t$.
- The power transmitted by the reader, is limited to 4W according the regulations presented in subsection 2.3.3. Therefore, $P_{tr}G_{tr} = 4\,\text{W}$, expressed in watts, or $P_{tr}\,(\text{dBW}) + G_{tr}\,(\text{dBW}) = 6\,\text{dBW}$, expressed in decibels.
- We assume the gain of the tag's receiver antenna is equal to one, so $g_{rt} = 1$ or $g_{rt}\,(\text{dBW}) = 0\,\text{dBW}$.

- Given we will use a circular polarized reader antenna with linearly polarized tag, the sensitivity to polarization can be ignored but an additional 3 dB loss has to be considered [70]. Therefore, $\chi\,(\mathrm{dBW}) = 3\,\mathrm{dBW}$.
- We make the assumption there are no losses due to the impedance match, $\tau = 1$ or $\tau\,(\mathrm{dBW}) = 0\,\mathrm{dBW}$.
- We neglect the large-scale and small-scale fading modeling, as it is too complex for the scope of this Thesis.

According to the previous reasoning, the unknown variable is the path loss, which directly depends on the distance between the reader and the tag. As presented in subsubsection 2.3.1.2, we can consider two different models for our case. On the one hand, the free-space wireless channel path loss, Equation 2.3, where the distance is

$$d = \frac{\lambda}{4\pi\sqrt{PL}} \tag{3.10}$$

The second model is the indoor large scale propagation wireless channel model, presented in Equation 2.4, in which we set the arbitrary distance d_0 to be one meter and consider a path loss exponent $n = 2$ or $n = 3$ and the Gaussian random variable $X_\sigma = 0$. Then, distance is computed as follows

$$d = d_0 \cdot 10^{\frac{PL(d) - PL(d_0)}{10n}} \tag{3.11}$$

Finally, the tag's power sensitivity S_t depends on the tag's integrated circuit, which is an architectural decision of our system. The different power sensitivities for the *Higgs-3*, *Higgs-4* and *Higgs-EC* ICs are -20 dBm, -20.5 dBm and -22.5 dBm, respectively (see section D.4). In Table 3.4 we present the maximum reading distances obtained when applying Equation 3.9 with Equation 3.10 and Equation 3.11, respectively. In all models a carrier frequency of 915 MHz is considered.

Tag IC	Propagation Model		
	Free-space	**Indoor LS, $n = 2$**	**Indoor LS, $n = 3$**
Higgs-3	11.65 m	11.65 m	5.88 m
Higgs-4	12.35 m	12.35 m	6.11 m
Higgs-EC	15.54 m	15.54 m	7.13 m

Table 3.4: Maximum reading distances for each tag IC and wireless propagation model.

The first two models turn out to provide same results. That is because when using $n = 2$ in the indoor large scale model we are actually considering a free-space model. Then, when using the indoor large scale model with $n = 3$ we obtain notably low values for the maximum distance, which differ from the results in [59]. Given all architectures will share the same propagation model, we consider using the free-space model is a correct approximation for our case.

3.3.2 Coverage

In subsection 3.3.1 we showed how the different tag technologies affect the effective communication range between the reader and the tag and how the tag's IC is key in this result. In previous sections we introduced the idea of achieving omnidirectional coverage, in which we have maximum communication distance or range is constant throughout the 360° in the horizontal plane.

Although RFID omnidirectional antennas are being studied [96][97], commercial RFID antennas present a directional gain profile, as showed in Figure 2.13. Via the *Antennas per reader* decision, we try to sectorize the terminal in order to have omnidirectional coverage. At this point two questions arise: How much the omnidirectional range varies depending on the number of antennas connected? And what effects does the selected RFID antenna have on the profile?

We start by answering the latter. All the RFID antennas available at the supplier's catalog (see section D.2) fulfill the basic specifications regarding RFID technology: Allowed UHF frequency bands, circular polarization, admissible size, resistance to temperature and humidity... However, there is one feature which is different within the antennas and is key for our purpose: The 3 dB beamwidth.

The 3 dB beamwidth is the angle that forms between the maximum gain direction and the direction whose gain is 3 dB below the maximum. It stands as a measure for the directivity of an antenna; highly directive antennas will present a low 3 dB beamwidth whereas antennas closer to omnidirectionality will have larger values. Given that a low-directivity antenna would be better for our case, we have selected ALR-A0501 antenna, as it is the one with the highest 3 dB beamwidth, being 105°. Figure 3.9 shows its radiation plot.

Our goal is to place more than one of these antennas in the same spot, connecting them to the same reader, and achieve omnidirectional coverage. In the morphological matrix, the *number of antennas per reader* decision has a direct impact on the desired omnidirectional coverage. In Figure 3.10, Figure 3.11 and Figure 3.12 we show the approximate radiation plots that would result from the connection of two, three and four ALR-A0501 antennas, respectively.

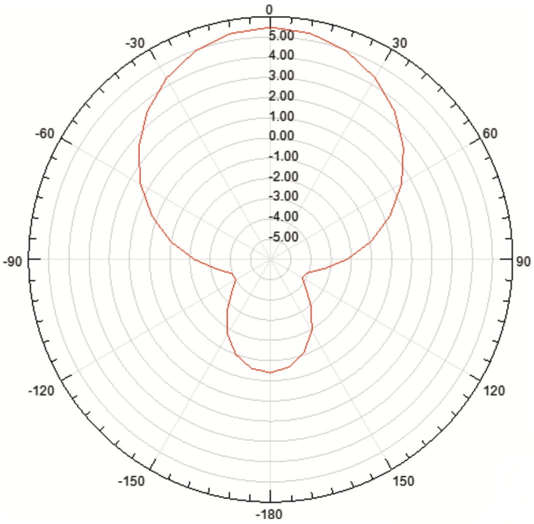


Figure 3.9: ALR-A0501 antenna radiation plot.

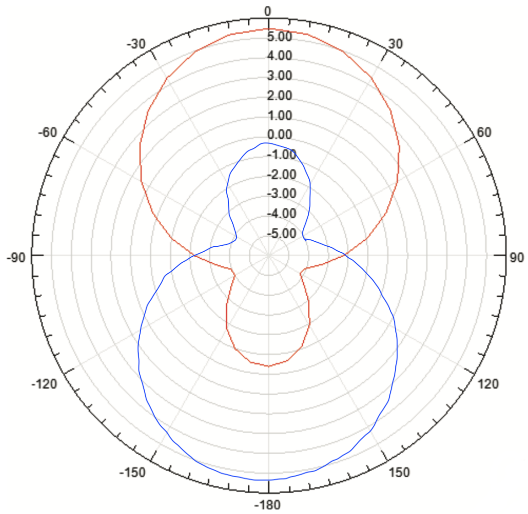


Figure 3.10: Two antennas radiation plot.

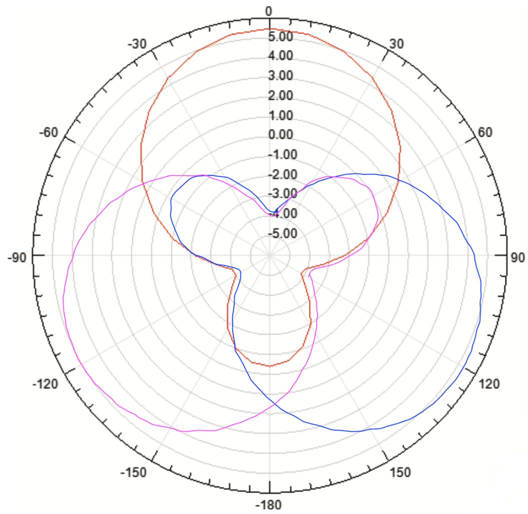


Figure 3.11: Three antennas radiation plot.

We can immediately appreciate how each of these radiation plots differ one to another in terms of omnidirectionality, as the more antennas we connect, the more "omnidirectional" the coverage becomes. At this point a critical analysis is necessary, as the radiation plot for the two antennas case in Figure 3.10 resembles more to an ellipse rather than a perfect circle. If we were to model the placement of this coverage, not only we would have to decide where to place the reader, but also in which direction we place the major axis of the ellipse, thus introducing a new variable to the system. We remove this option from the APR decision, as it does not fulfill our goal of achieving omnidirectional coverage. This new refinement is shown in Table 3.5.

The options that consider three and four antennas show a radiation plot close to an

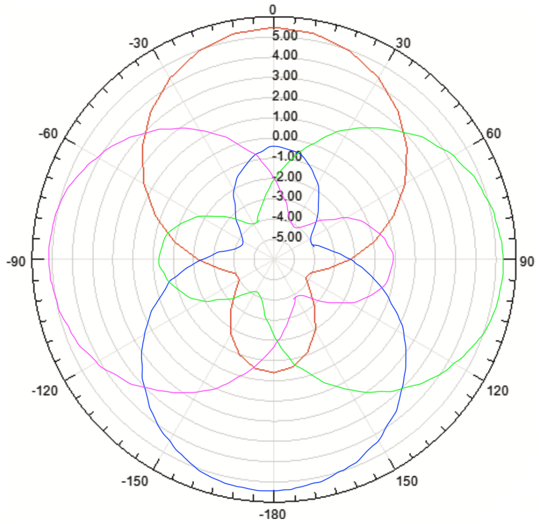


FIGURE 3.12: Four antennas radiation plot.

Architectural Decision	**Option 1**	**Option 2**	**Option 3**	**Option 4**
Antennas per Reader (APR)	3	4		

TABLE 3.5: Redefinition of the *Antennas per Reader* decision.

omnidirectional, making it clearer in the latter case. The next step is to calculate the extent of the coverage of both antenna settings. We have already computed the maximum range for the maximum gain direction but we can not assume the omnidirectional coverage range will equal the maximum range. We have to introduce a correction factor as we do not have perfect omnidirectional coverage.

For the three antennas case, we realize the 3 dB beamwidth of the three antennas covers almost the 90% of the azimuth plane, which means the range of the omnidirectional coverage is the maximum range with a 3 dB penalization. Therefore, the power sensitivity values to take into account for *Higgs-3*, *Higgs-4* and *Higgs-EC* would be -17 dBm, -17.5 dBm and -19.5 dBm, respectively.

On the other hand, for the four antennas case, the azimuth plane is entirely covered by the 1.5 dB beamwidth. So, we penalize the power sensitivity with 1.5 dB extra loss, as it is the difference between the maximum gain of the antenna and the gain of the 45º direction. As a result, the power sensitivity values to take into account for *Higgs-3*, *Higgs-4* and *Higgs-EC* would be -18.5 dBm, -19 dBm and -21 dBm, respectively.

In Table 3.6 we show the resulting coverage ranges for each IC depending on the number of antennas connected to the reader.

Tag IC	Number of antennas	
	Three (3 dB loss)	Four (1.5 dB loss)
Higgs-3	8.25 m	9.81 m
Higgs-4	8.74 m	10.39 m
Higgs-EC	11.00 m	13.08 m

TABLE 3.6: Coverage ranges for each tag IC given a number of connected antennas.

3.3.3 Reader Placement

Once we have quantified the coverage range we are ready to discuss the placement of the readers within the terminal. To avoid misleading during the following discussion, in Figure 3.13 we show, as an illustrative example, a basic placement of the RFID readers, so the reader can get a grasp of how the coverage of the dense reader environment is modeled. The circumferences represent the each of the readers' coverage range, and all tags inside a reader's coverage can be reached by that reader.

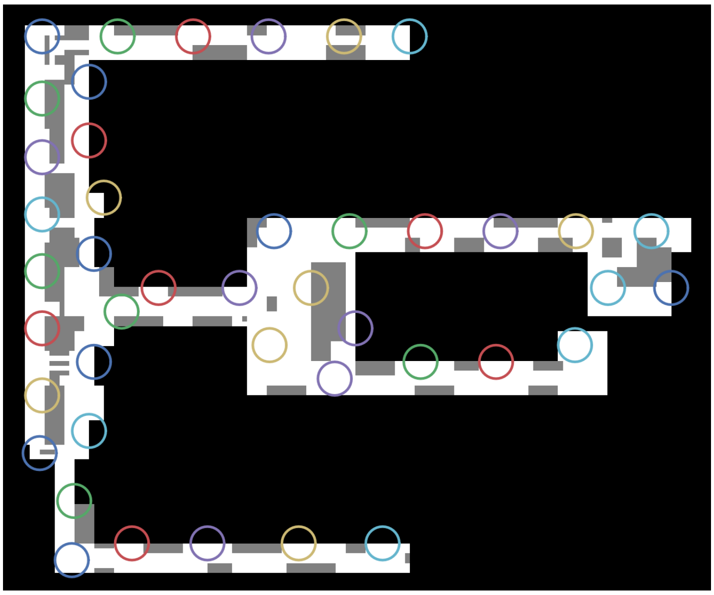


FIGURE 3.13: Illustrative example of a simple placement of readers within the modeled terminal.

In previous sections we introduced the necessity of developing a methodology to place the different readers within the terminal, as placing too many readers in a determined area may lead to undesired interchannel interference.

This problem is a clear example of the Antenna Placement Problem (APP), a subtype of the "covering problems" [98] which has already been addressed for the RFID case [99]. The APP poses the question of how many and where we should place different base stations given a set of candidate points and a set of constraints.

The most common solution for the problem consists in the application of what are known as genetic algorithms [100], a type of optimization algorithms inspired by natural selection. Instead of examining the performance of each solution, these algorithms start by considering just a set of all the solutions and determining which of the solutions from the set present the best performance. From these, they generate new solutions by selection, crossover and mutation of these solutions' features, creating a new set of improved solutions.

The application of genetic algorithms to solve the APP has already been studied in literature [101]. Unfortunately, when the number of candidate points increases, it takes more time to converge to the optimal solution. In our case, where we have more than a thousand candidate points, and more than a hundred architectures, each with its own APP, the use of genetic algorithms has to be discarded due to the time expense each run would take.

Instead, we develop a suboptimal placement algorithm all architectures share and fulfills the channel interference restrictions while being computationally inexpensive. We change the *Reader Placement Heuristic* decision to *Horizontal Distance Between Readers* (HDBR). Our goal is to find which should be the minimum horizontal separation between readers -fixing the vertical one- to avoid the interference. We start by considering a separation of exactly the readers' coverage, as in the first diagram in Figure 3.14. The location of the second reader is tangential to the first reader's coverage.

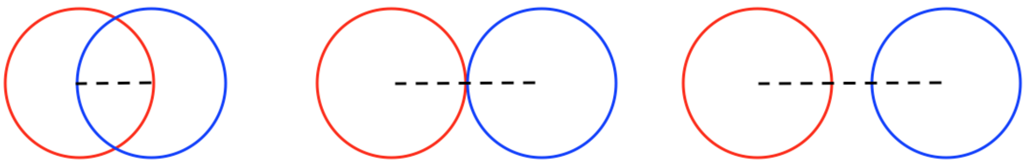


FIGURE 3.14: Examples of different horizontal distances between readers.

This first approach, although it covers the entire terminal, fails to adequately allocate all frequency channels without creating interference. Following the second diagram of Figure 3.14, we now increase the distance to two times the coverage radius. This solution fails to allocate channels in the spacious areas of the terminal such as the nexus of concourses H and K.

At this point we realize we have to increase the horizontal distance between readers to a bigger amount than the coverage diameter. Therefore, we focus on the third diagram in Figure 3.14. To obtain this minimum distance we develop two algorithms that determine the correctness of a certain placement. Both algorithms have also been used to discard the two options mentioned in the previous paragraph.

First, to evaluate a certain placement of all readers in terms of the ability to correctly allocate all communication channels, we develop algorithm 3, a recursive algorithm that tries to converge to a solution in which each reader can communicate through a channel without interference.

Algorithm 3 Obtain Channel Allocation

```
procedure ITERATE_READERS(readers_array, i)
    if i == length (readers_array) then
        return True
    else
        for c ∈ [1, 50] do
            if check_channels (readers_array, i, c) then
                readers_array[i].channel ← c
                if iterate_readers (readers_array, i + 1) then
                    return True
                readers_array[i].channel ← −1
        return False
```

The algorithm iterates over all the readers placed in the terminal and starts allocating channels examining the viability of the current allocation at each step of the iterative process. If a new reader can not communicate through any channel without creating interference, the algorithm changes the allocation of the previous readers, until it reaches convergence. To check the correctness of the placement at each step we develop algorithm 4.

Algorithm 4 Check Channel Interference

```
procedure CHECK_CHANNELS(readers_array, i, c)
    for r ∈ readers_array[: i − 1] do
        if check_line_of_sight (readers_array[i], r) then
            distance ← compute_distance (readers_array[i], r)
            channel_difference ← abs (c − r.channel)
            if channel_difference == 0 & distance ≤ 1400 then
                return False
            if channel_difference == 1 & distance ≤ 180 then
                return False
            if channel_difference == 2 & distance ≤ 130 then
                return False
            if channel_difference == 3 & distance ≤ 95 then
                returnn False
    return True
```

The algorithm focuses on reader i and iterates over all the other readers that have been assigned a channel. First, it checks whether reader i has direct line of sight (LOS) with each of the other readers. We not only consider as LOS a clear path between readers but we also, trying to be conservative, allow an obstacle as thick as 8 meters. So, if a reader is in LOS with reader i, we compute the distance between readers and the distance between the readers' assigned channels. We then compare both measures with the interference-safe values from Table 2.6, considering both readers have face-to-face antennas and therefore considering the most restricted case, being the Front Configuration.

We apply the algorithms to increasing values for the new HDBR decision and find the minimum distance for the convergence of the algorithms is 2.5 times the coverage radius. This value is set to be the first option for the HDBR decision while the other options must be greater values than 2.5. Therefore, we select 2.5, 2.75, 3 and 3.25 times the radius as the horizontal distances between readers.

3.3.4 Communication Protocol

The final step of the involves the design of algorithms that take care of the data link layer communications. As introduced in subsection 2.3.2, RFID communication employs a time-based MAC protocol, mostly the Slotted ALOHA protocol. The protocol works as follows:

- There is a reader and a set of n unidentified tags.
- The reader starts the communication process by providing N time slots and broadcasting an identification cycle to all tags in range.
- The tags receive the number of slots N and randomly select one to communicate their IDs back to the reader.
- Given tags do not communicate with each other, the reader receive some slots empty, others with one tag's ID, and the rest are corrupted due to data collisions from two or more tags.
- The reader triggers more cycles until it estimates it has identified all tags in range. Once a tag is identified for the first time, it will remain identified during the following cycles for the rest of the set of cycles.
- Once it finishes with the identification process, all tags go unidentified again and the reader has to restart the complete set of cycles.

3.3.4.1 Mathematical Preliminaries

Following the work in [8] to create our model, we start by introducing some mathematical preliminaries. Given a reader R, which provides N slots to the set of n tags that it covers, the number r of tags in one slot —called occupancy number— follows a binomial distribution with parameters n and $\frac{1}{N}$ [8]:

$$B_{n,\frac{1}{N}}(r) = \binom{n}{r}\left(\frac{1}{N}\right)^r\left(1-\frac{1}{N}\right)^{n-r} \tag{3.12}$$

Applying Equation 3.12 to all N slots, the expected value of the number of slots with occupancy number r is given by $a_r^{N,n}$ [8]:

$$a_r^{N,n} = N\,B_{n,\frac{1}{N}}(r) = N\binom{n}{r}\left(\frac{1}{N}\right)^r\left(1-\frac{1}{N}\right)^{n-r} \tag{3.13}$$

According to the description of the reading process at the beginning of this section, during a set of reading cycles, the number of identified tags at the end of a cycle is never less than the number of identified tags at the end of the previous cycle, as once a tag is identified during a set of cycles, it continues to be so until the end of the cycle. This process can be viewed as a Markov process, where the state of the reader is the number of tags identified at a given time. This way, for a set of n tags, the reader has $n+1$ different states. At a certain time, if the reader is in state $k_t \in (0, 1, \cdots, n)$, the next state will be k_{t+1}, with $k_t \leq k_{t+1} \leq n$. The Markov chain for this process is showed in Figure 3.15.

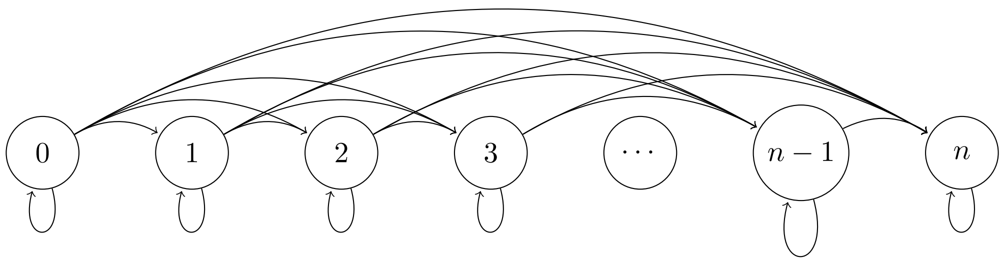


FIGURE 3.15: Tag identification viewed as a Markov chain.

In this process, the probabilities for changing from state i to state j are given by q_{ij}, which is defined as follows [8]:

$$q_{ij} = \begin{cases} 0 & \text{if } j < i \\ \sum_{r=0}^{i} P(\mu_1 = r)\frac{\binom{i}{r}}{\binom{n}{r}} & \text{if } j = i \\ \sum_{r=j-i}^{n} P(\mu_1 = r)\frac{\binom{n-i}{j-i}\binom{i}{r-j+i}}{\binom{n}{r}} & \text{if } j > i \end{cases} \tag{3.14}$$

$P(\mu_r = m_r)$ denotes the probability of having m_r slots filled with exactly r tags and adopts the following expression [8]:

$$P(\mu_r = m_r) = \frac{\binom{N}{m_r}\prod_{k=0}^{m_r-1}\binom{n-kr}{r}G(N-m_r, n-rm_r)}{N^n} \tag{3.15}$$

Where [8]

$$G(M,m) = M^m + \sum_{k=1}^{\lfloor \frac{m}{r} \rfloor}\left\{(-1)^k \prod_{j=0}^{k-1}\left\{\binom{m-jr}{r}(M-j)\right\}(M-k)^{m-kr}\frac{1}{k!}\right\} \tag{3.16}$$

The equations presented in this subsection conform the theoretical base of the S-ALOHA protocol, and will be used as reference for our model.

3.3.4.2 Model Implementation

Slotted ALOHA allows some flexibility within its implementation to be able to serve different types of applications that require a diverse range of slot frame sizes, throughput, or reading speed. One of the simplest algorithms is the Basic Framed Slotted ALOHA (BFSA) algorithm, which although easy to implement, causes inefficiencies when selecting optimal slot frame sizes [102]. In [8], the Dynamic Framed Slotted ALOHA (DFSA) is presented, and its Advanced (AFSA) version is introduced too. Both algorithms try to estimate the number of tags present in the area before deciding the frame size. Finally, in [102] the Enhanced Dynamic Framed Slotted ALOHA (EDFSA) is presented and it is shown how it notably outperforms BFSA in terms of frame size efficiency. When the number of tags exceeds 500, we can also appreciate how EDFSA provides a more slot-efficient approach than DFSA does. Accordingly to the scope of our problem, given we will not have numbers as high as 500 of tags within readers' coverage areas, we choose to model the DFSA/AFSA algorithm, as, for our case's tag quantity, performs as well as EDSFA and is less complex.

DFSA algorithm estimates the number of tags present in the reader's radius before selecting an optimal frame size. To that end, the reader first provides a frame size of N slots. The received frame, result of a tag-reading cycle, can be viewed as a triple of numbers $\langle c_0, c_1, c_k \rangle$, in which each number quantifies the number of empty slots, slots filled with exactly one tag and slots with collisions.

Given these three numbers we can estimate a lower bound for the number of tags n, given by the following expression [8]:

$$\varepsilon_{lb}(N, c_0, c_1, c_k) = c_1 + 2c_k \tag{3.17}$$

This expression meets the requirement that there can not be less tags than the number of slots filled with one tag plus two times the number of slots in which a collision has taken place. A collision requires a minimum of two tags filling the same slot in order to occur, that is why c_k is multiplied by two.

Once the reader knows a lower bound for the number of tags n, it can compute a more accurate estimation given by the next equation [8]:

$$\varepsilon_{vd}(N, c_0, c_1, c_k) = \min_{n} \left| \begin{pmatrix} a_0^{N,n} \\ a_1^{N,n} \\ a_{\geq 2}^{N,n} \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ c_k \end{pmatrix} \right| \tag{3.18}$$

The reader looks for the value for n that minimizes the error between the expected numbers of slots filled with zero, one or two or more tags, given by Equation 3.13, and the actual quantities for the different cases. Algorithm 5 models the procedure explained above, which is performed by each of the readers repeatedly.

Algorithm 5 Request IDs of tags in range

```
procedure REQUEST_IDs(estimate)
    detected ← {}
    undetected ← all_passengers_in_range
    for cycle ∈ [1, num_cycles] do
        empty, one, colli ← broadcast (detected, undetected)
        if cycle == 1 and estimate == True then
            n_est = estimate_n (empty, one, colli)
            N = ⌊n_est · e⌋ + 1
```

The algorithm starts by assuming no passenger in range is identified and starts performing identification cycles, as many as given by decision *Number of ALOHA cycles*. At every cycle, the reader updates its information on identified and unidentified passengers and, if the cycle is the first of the set, estimates the number of passengers/tags n in area using Equation 3.17 and Equation 3.18. Then, updates the size of the slot frame accordingly to the expression $N = \lfloor n\_est \cdot e \rfloor + 1$. The reason of this number is that given a number of agents k following a Slotted ALOHA MAC protocol, the frame size that provides the optimal throughput is $N = k \cdot e$ [103]. Finally, the broadcast function is what manages the status of identified and unidentified passengers inside a reader's coverage area. It is shown as algorithm 6.

The algorithm first makes each tag in range select a random slot to communicate. Then checks the total number of tags that have filled each slot and, among those slots that are filled with just one tag, looks for that tag's ID to check if it is already within the detected group. If not, it is added to the detected group and deleted from the undetected group.

Algorithm 6 Broadcast an ID Request to all tags in range

```
procedure BROADCAST(detected,undetected)
    vector ← {}
    empty ← N
    one ← 0
    colli ← 0
    for p ∈ {all_passengers_in_range} do
        selected_slot ← p.select_slot(N)
        if selected_slot ∈ vector then
            if vector[selected_slot] ! = −1 then
                vector[selected_slot] ← −1
                one ← one − 1
                colli ← colli + 1
        else
            vector[selected_slot] ← p
            empty ← empty − 1
            one ← one + 1
    for number ∈ {vector} do
        if vector[number] ! = −1 then
            if vector[number].ID ∈ {undetected} then
                detected[vector[number].ID] ← vector[number]
                delete undetected[vector[number].ID]
    return empty, one, colli
```

At the beginning of a set of cycles, the undetected group is formed by all passengers in range whereas the detected group is empty. At the end of the set of cycles, hopefully the situation is the inverse, but it can happen there are some tags left in the undetected group, which means the passengers carrying those tags will not be tracked during that set of cycles.

3.4 Metrics

The final step of the model is the selection of evaluation criteria in order to compare the different architectures and be able to discern which of them provides the best results. To that end, in this section we focus on setting a pair of metrics to allocate the different architectures within an architecture tradespace.

Ideally, we would like to point out which architecture is the one that provides the greatest value, defined as benefit at cost. In order to measure the benefit and the cost separately we create both a performance metric and a cost metric, and assign each architecture two individual values that allow the placement of that architecture in the architecture tradespace.

Before going any further, in Table 3.7 we present the current state of the morphological matrix, given it has been redefined during the analysis executed in the previous section.

Architectural Decision	**Option 1**	**Option 2**	**Option 3**	**Option 4**
Tracking Departing Passengers (TDP)	Yes	No		
Tracking Arriving Passengers (TAP)	Yes	No		
Tracking Connecting Passengers (TCP)	No			
Departing Initial Tracking Point (DITP)	Entrance	Check-in Counter	Security Point	None
Departing Final Tracking Point (DFTP)	Check-in Counter	Security Point	Gate	None
Antennas per Reader (APR)	3	4		
Tag Technology (TT)	*Higgs-3*	*Higgs-4*	*Higgs-EC*	
Horizontal Distance Between Readers (HDBR)	2.50 radius	2.75 radius	3.00 radius	3.25 radius
Number of ALOHA Cycles (NOAC)	2	3	4	

Table 3.7: Refinement of the morphological matrix after finishing the model of the system.

3.4.1 Cost Metrics

The first of the metrics deals with the cost of the architectures. There are different units with which we can measure the cost of the system, depending on the architect's intentions and the system's decisions. For instance, we could use the number of placed readers as the cost unit, but we would be missing the impact of the *Antennas per Reader* or *Tag Technology* decisions.

In order to use a fair metric that takes into account all the costs involved in the system fairly, we use US Dollars as the cost unit. The reason behind this is that, ultimately, all the costs that derive from the implementation of this system can be translated to economical costs. So, we divide the costs involved in the architectures in the following three categories:

- Cost of the technology.
- Cost of the installation.

- Cost of the operation.

We will analyze each of the above-mentioned costs separately.

3.4.1.1 Cost of the Operation

The operating costs this system involves are related with, on the one hand, the human labour cost of managing and monitoring the system and, on the other hand, the cost of having the system working, i.e., the cost of the power consumed.

The cost of the human operating of the system is inevitably intrinsic to all architectures, as the operator controlling the system will be dealing with the same computer interface independently of the number of readers installed. Therefore, it is fair to neglect this cost in the metric, as otherwise it would mean adding the same quantity in the cost metric for all architectures.

In contrast, the cost of the power consumed is directly proportional to the number of readers placed in the terminal. However, in section 2.3 it was mentioned that passive RFID systems are relatively inexpensive to operate, compared to the cost of the hardware. So, maybe considering the cost of the power consumed would not provide useful information in the global cost metric but add more complexity to the model. In order to decide whether we should include it or not, we model the cost of the power consumed C_{power} as

$$C_{power} = \sum_{i=1}^{N_R} \sum_{j=1}^{N_C} P_c\{\text{W}\} \cdot s_{ij}\{\text{slots}\} \cdot f_1 \left\{\frac{\text{s}}{\text{slots}}\right\} \cdot \frac{1\text{h}}{3600\,\text{s}} \cdot \frac{1\text{kW}}{1000\text{W}} \cdot f_2 \left\{\frac{\$}{\text{kWh}}\right\} \quad (3.19)$$

Where P_c is the power consumed by all readers in watts, s_{ij} is the number of slots provided by the reader i in the cycle j, f_1 is a factor that converts slots provided to seconds, based on Table 2.7, and f_2 is a second factor that models the price of the kWh. For our case, according to the RFID readers datasheets, we choose 13 watts as the power consumed, and set the price of the electricity to $0.12/kWh, according to the US average.

After defining Equation 3.19, in order to determine the necessity of including the cost of the power consumed in our cost metrics, we set the architecture tradespace just for the *Number of ALOHA cycles* decision, as showed in Figure 3.16.

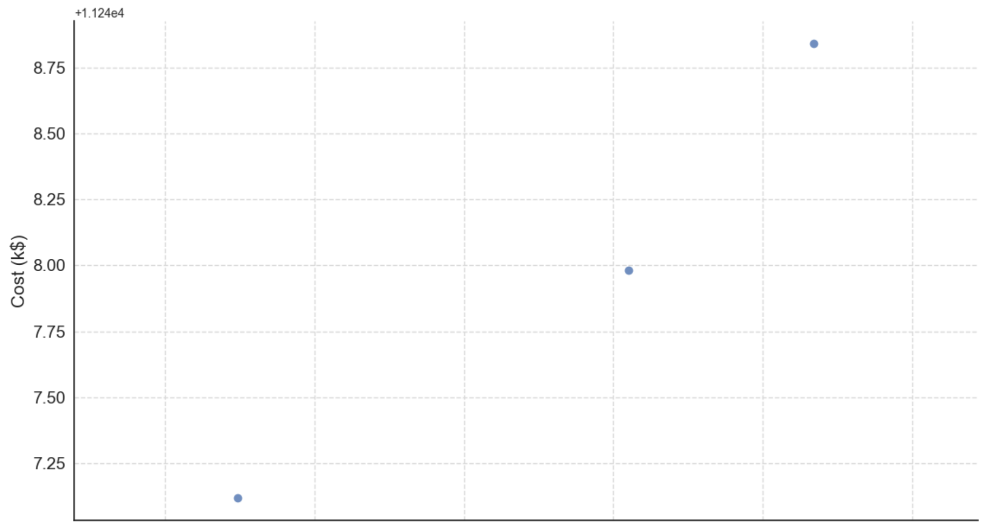


FIGURE 3.16: Architecture tradespace just considering the *Number of ALOHA cycles* decision.

As the reader may have noticed, the difference in the cost between the different architectures is derisory in comparison to their actual nominal cost. So, we can neglect the power consumption cost from the global cost metric, as the information provided is not valuable and only adds complexity to the simulation model. Regarding the blind *Performance* axis, we will cover the performance metric in detail in the following section.

This result in the cost metrics makes us question the necessity of taking into account the *Number of ALOHA cycles* decision in further simulations, given now we can just select the option that performs the best, as we have shown this decision is not relevant to the cost metric. So, now one could think we select 4 cycles as the option for this decision and carry on, but we propose an alternative solution.

Instead of setting a limit for the maximum number of cycles, we set a limit on the duration of a set of cycles. Given the minimum coverage radius in Table 3.6 is 8.25 meters, we propose a maximum cycle set duration of five seconds. This way, given the average human walking speed is 1.5 m/s, we make sure a passenger will be able to be detected at least once in the worst-case coverage.

Within this cycle set duration of five seconds, we let the readers perform as many cycles as possible in a temporal frame size of four seconds, and establish a safe margin of one second to perform the other internal protocol operations of the s-ALOHA protocol —resetting all tags, creating the communication frame, overhead...—.

3.4.1.2 Cost of the Technology

The technology involved in the system consists basically in three different elements: RFID readers, antennas and RFID tags. As we have selected commercial models for the simulation model, we use the actual cost of these models to come up with the cost metrics. Accordingly, we contact the sales department of the supplier company and ask for the pricing of the different items.

By the time this thesis is being carried out, the supplier has two different 4-port RFID readers in its catalog: The ALR-F800 and the ALR-9680 (see section D.1). The sales responsible recommended considering ALR-F800 for the calculations, as it outperforms ALR-9680. Therefore we select the ALR-F800 model, which has a cost of $1,399.

Regarding to the antenna cost, in subsection 3.3.2 we chose the ALR-A0501 antenna for our calculations and modelings. This antenna has a price of $139, notably less than the RFID reader.

Finally, the cost of the tag depends on its integrated circuit. However, we first must choose a tag model that meet the requirements of our system and hopefully is available with each type of IC. In section D.3 we can observe the *Squiggle* family is the one that serves best to general purpose situations, as well as retail and mobile asset tracking. Among the models contained in the family, we choose the *Squiggle*(SQ) model, as it is available with the three types of IC and outperforms the *Squiglette*(ST) model. The pricing for each model, depending on the IC and the inlay, is shown in Table 3.8.

Tag IC	**Type of inlay**	
	White Wet	**Clear Wet**
Higgs-3, ALN-9640	$0.131	$0.121
Higgs-4, ALN-9740	$0.127	$0.116
Higgs-EC, ALN-9840	$0.124	$0.113

Table 3.8: Pricing of an RFID tag depending on its integrated circuit and its inlay.

At this point the reader may be surprised with the numbers in Table 3.8, as the IC *Higgs-EC*, which is the one that performs the best in coverage range, is also the cheapest. According to the company this is due to several reasons. First, the size of the *Higgs-EC* IC is smaller than the size of the other ICs, meaning the material cost is also reduced. Then, the manufacturing system has been improved and this is reflected in a reduction of the manufacturing costs. Finally, there exists a motivation from RFID to encourage costumers to use the newest tags available and stop using old models. All of these suggests a modification of the *Tag Technology* decision, implying the direct selection of the *Higgs-EC* option.

Finally, we can appreciate how small the cost of a single tag is compared to a cost of a reader on an antenna. Such a low cost could be charged to the passenger without him/her even noticing. This, alongside the fact that RFID tags have a warranty of six months —according to the company—, arising the possibility of tag reuse, makes us neglect the cost of the RFID tags and only consider the cost of the RFID readers and the antennas.

3.4.1.3 Cost of the Installation

Finally, the RFID system also implies installation costs, which can be divided in, on the one hand, the human labour and the infrastructure needed to correctly install and start the system; and on the other hand, the auxiliary materials and hardware that accompany the readers and connects each node in the network.

With the former cost we have a similar situation as the one regarding the cost of the human controlling of the system, where this cost is shared among all architectures and therefore can be neglected.

On the other hand, the cost of the additional hardware depends on the extension of the installation, as a larger number of readers suppose a larger amount and deployment of cabling. It is true that independently of the number of readers, all architectures will present a central processing unit that will take care of the data gathering from all readers. However, the cabling needed to communicate the readers may affect the cost of the system.

Consulting again with the sales department at the supplier company, we know the pricing of the cables for the antennas ranges from $30 (3 meters) to $40 (6 meters). Considering that, from one architecture to another, this cost would suppose as much as $30 (given the difference in the number of antennas per reader between architectures is one antenna as maximum and 3 meters is enough cabling), we opt for disregarding this cost. Also, we have to take into account the cables that connect the readers to the Local Area Network (LAN), which are commonly called Ethernet cables. Ethernet cables, thanks to the improvement in the manufacturing process, are known to be one of the cheapest cables to acquire and would not have a substantial impact on the cost metrics.

3.4.1.4 Metric

After carefully examining all the costs derived from the installation, activation and functioning of the RFID system, the cost involved C_{total} is function of

$$C_{total} = C_{operation} + C_{technology} + C_{installation} \approx C_{technology} \tag{3.20}$$

At the same time, the cost of the technology $C_{technology}$ can be decomposed as follows

$$C_{technology} = C_{readers} + C_{antennas} \tag{3.21}$$

3.4.2 Performance Metrics

During the process of setting the cost metrics we have continued with the iterative process of redefining the morphological matrix while analyzing how the different decisions affect the model. The last step of the description of the model is coming up with appropriate performance metrics based on the morphological matrix. Therefore, we will consider the current morphological matrix, shown in Table 3.9, as the definitive one for our system.

Architectural Decision	**Option 1**	**Option 2**	**Option 3**	**Option 4**
Tracking Departing Passengers (TDP)	Yes	No		
Tracking Arriving Passengers (TAP)	Yes	No		
Tracking Connecting Passengers (TCP)	No			
Departing Initial Tracking Point (DITP)	Entrance	Check-in Counter	Security Point	None
Departing Final Tracking Point (DFTP)	Check-in Counter	Security Point	Gate	None
Antennas per Reader (APR)	3	4		
Tag Technology (TT)	*Higgs-EC*			
Horizontal Distance Between Readers (HDBR)	2.50 radius 1	2.75 radius	3.00 radius	3.25 radius

TABLE 3.9: Definitive morphological matrix.

At this point, given the constraints that were presented in Table 2.10, the number of viable architectures is 144. We can appreciate how we have downsized the matrix during this chapter and thus reduced the complexity of our model. One of the main changes is the fact that now all formal decisions of the matrix relate to the coverage of passengers rather than their tracking, as we have come up with an alternative way of regarding the ALOHA protocol in the previous section. Therefore, it is reasonable to focus the performance metric on the coverage too.

In subsubsection 3.2.2.2 we defined a passenger's trajectory as $PT_l = (s_{l1}, s_{l2}, \cdots, s_{lK})$, where each s_{lk} corresponds to a step of the trajectory that has associated certain coordinates i_{lk}, j_{lk}. Given a trajectory PT_l we now define its coverage C_l as

$$C_l = (C_{l1}, C_{l2}, \cdots, C_{lK}) \tag{3.22}$$

Where each C_{lk} is either 1, if the step s_{lk} is covered by a reader, or 0, if it is not. So, as the main performance metric, we define the percentage of spatial covered trajectory P_C as

$$P_C = \frac{1}{N_P} \sum_{l=1}^{N_P} \frac{1}{N_{lK}} \sum_{k=1}^{N_{lK}} C_{lk} \tag{3.23}$$

Where N_P is the total number of passengers that get in the terminal during the whole simulation time frame. We define it as "spatial" because we consider the amount of a passenger's path that is covered and not the amount of time a passenger is being covered.

Although it is not affected by the decisions, given that after the covering function the system has to actually detect the passengers, we develop a second metric that does deal with the tracking and will be used to further evaluate the architectures. During its trajectory, each passenger l's tag will be requested several times. We define this requests R_l as

$$R_l = (r_{l1}, r_{l2}, \cdots, r_{lQ}) \tag{3.24}$$

Where r_{lq} is either 1, if the tag is correctly identified in time q, or 0 if it is not. Then, the second performance metric P_T, dealing with the tracking is defined as follows

$$P_T = \frac{1}{N_P} \sum_{l=1}^{N_P} \frac{1}{N_{lQ}} \sum_{q=1}^{N_{lQ}} r_{lq} \tag{3.25}$$

With this second metric, we have two different metrics, each for one of the main functions of our system: Covering and tracking, but just the coverage metric is considered the global performance metric of our system. In the next chapter we will elaborate the architecture tradespace, which will allow us compare the different architectures and provide us insightful results.

Chapter 4

Results

All the different parts of the model described in the previous chapter serve to create a connection from the envisioned concept and architectural decisions to the architectural tradespace that compares them all. The objective of this chapter is to provide the tradespace and draw all the conclusions that can be extracted from it.

To that end, in section 4.1 we briefly present the data that is used to base the simulation on. Then, section 4.2 introduces the results of a baseline case and deepens on the impact of each architectural decision. In section 4.3 we discuss the effect different changes in the baseline model have on the tradespace, extracting further conclusions. Finally, section 4.4 presents a discussion on the best options for each of the decisions.

4.1 Data

One of the important parts of the model is the use of real data for the shared context of all architectures. Specifically, we are interested in modeling a real day of activity at Chicago airport; that is, creating a schedule of flights, each with its own set of passengers, its time of departure or arrival, and a gate at the terminal.

For this purpose we extract the data from the Chicago O'Hare Airport's flights schedule website. There are two separate websites, one for the departures [104] and one for the arrivals [105]. Each contains the necessary information for modeling a flight, with the time it arrives or departs, the gate of the flight at the terminal, and the type of plane.

However, the website only allows the access to the current day schedule, the day before, and the day after. This is why we choose dates within the time frame of this Thesis. As a baseline case, we model the complete flight schedule of Thursday, July 27$^{\text{th}}$ 2017.

That day, a total of 1,122 flights were scheduled, divided in 572 departures and 550 arrivals.

Based on this schedule, the simulation code creates all the passengers and their trajectories once so each architecture performs under the same conditions. This way, we ensure there will not be variability in the results from no input but the decisions of the system.

4.2 Baseline Results

There are different methods to solve a system architecture optimization problem in order to find the best architectures that meet the problem requirements. To that end, given the number of architectures in this case is relatively small (compared to the billions of architectures that may be considered in other system architecture problems) we follow a Full-Factorial Enumeration strategy and simulate each of the 144 architectures.

We present the architecture tradespace following a full-factorial enumeration on all the decisions in Figure 4.1. All architectures show both the coverage metric in the x-axis and the cost metric in the y-axis.

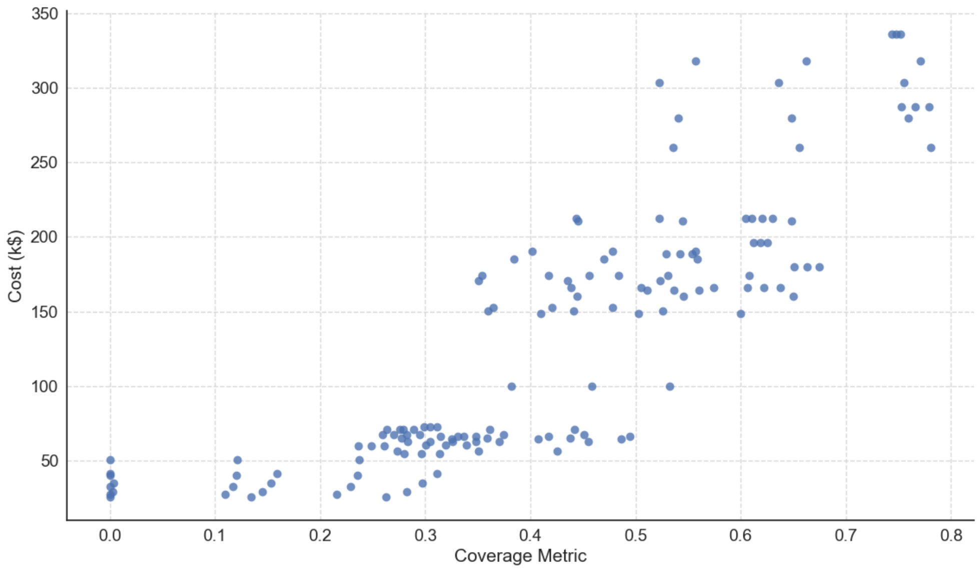


Figure 4.1: Architectural Tradespace for the baseline case.

At first glance, there exists a proportional relation between the performance and the cost of the system. So, if a coverage of more than a 70% is to be achieved, the cost of the system inevitably rises to more than 250 thousand dollars. In the same way, if for instance there is a budget limit of 120 thousand dollars, the maximum coverage that can be achieved is around a 53%. We define the *utopia point* of this tradespace to the ideal architecture that would provide a 100% coverage with zero cost. This point is unreachable, and it is task of the architect to determine which architecture/s serve better to the system's goal in absence of the utopia point architecture.

4.2.1 Non-Dominated Architectures

One of the first tasks to optimize a system architecture given a tradespace is to profile its Pareto front, the set of non-dominated architectures. An architecture A_i is said to be non-dominated if there is no other architecture A_j that outperforms it for the same or less cost. In other words, A_i will belong to the Pareto Front if

$$\forall j \neq i, \text{ if } C_{A_j} \leq C_{A_i} \text{ then } P_{A_i} \geq P_{A_j} \tag{4.1}$$

Where C_{A_k} and P_{A_k} are the cost and the performance values for architecture A_k, respectively. According to the previous definition, we identify all non-dominated architectures in Figure 4.1 and draw the Pareto Front accordingly, as shown in Figure 4.2.

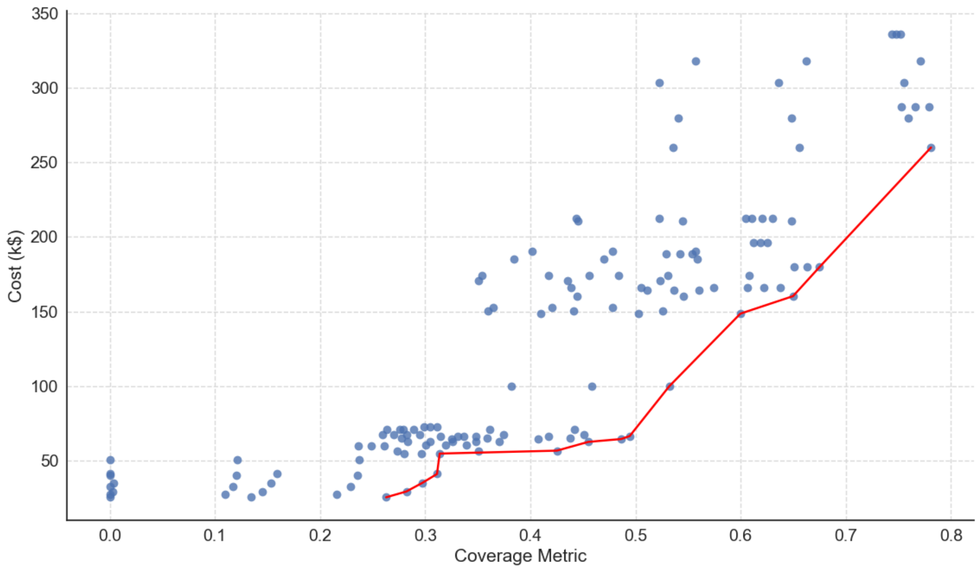


FIGURE 4.2: Architectural Tradespace and Pareto Front for the baseline case.

For the baseline case, the Pareto Front is composed by 14 architectures that span from the low-cost-low-performance to the high-cost-high-performance regions. These architectures are listed in Table F.1. Starting from the low-cost region, the Pareto Front presents a tendency such that an increase in performance comes alongside an increase in the cost of the architecture. Then, it changes and we enter a stage in which we can achieve substantial increases in performance with minimal increases in cost. Finally, the Pareto Front goes back to the initial tendency until it reaches the high-cost-high-performance area. From this, we can see how easily we can move across the range of 30%-50% in performance without noticing a significant change in cost.

4.2.2 Impact of the Architectural Decisions

After obtaining the Pareto Front, the next important step is to determine how the architectural decisions affects the tradespace, so we have more information towards the architecture optimization.

One of the first things we want to highlight is the gaps in cost present between different groups of architectures in the tradespace. For instance, it is unfeasible to have an architecture with a cost between 100 and 150 thousand dollars while there are more than thirty architectures ranging from 150 to 200 thousand dollars. This is caused by the *Departing Final Tracking Point* decision, as shown in Figure 4.3.

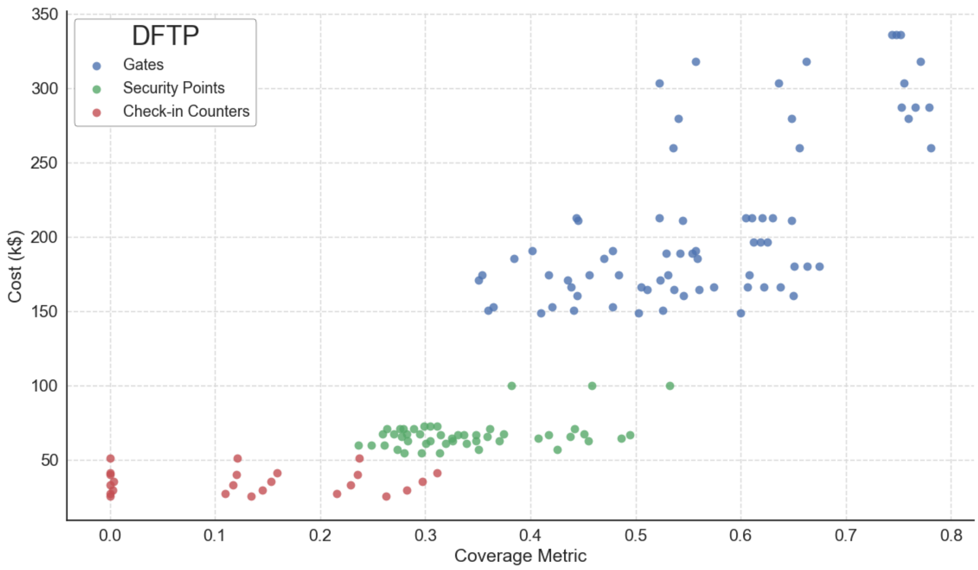


FIGURE 4.3: Impact of the *Departing Final Tracking Point* decision on the Architectural Tradespace for the baseline case.

In the O'Hare airport, most of the surface is comprised between the security points and the gates. Therefore, if readers are placed on that area (DFTP = Gates) the amount of coverage substantially increases as well as the number of readers needed to cover the surface according to the heuristic used. On the other hand, if readers are limited to the area between the entrance and the check-in counters, the maximum amount of possible surface to be covered is notably small (30%), therefore the little cost of these architectures.

Still, there is a cost gap between architectures that have the same option for the DFTP decision in the high-cost-high-performance region. This gap is explained by the *Horizontal Distance Between Readers* decision, as shown in Figure 4.4.

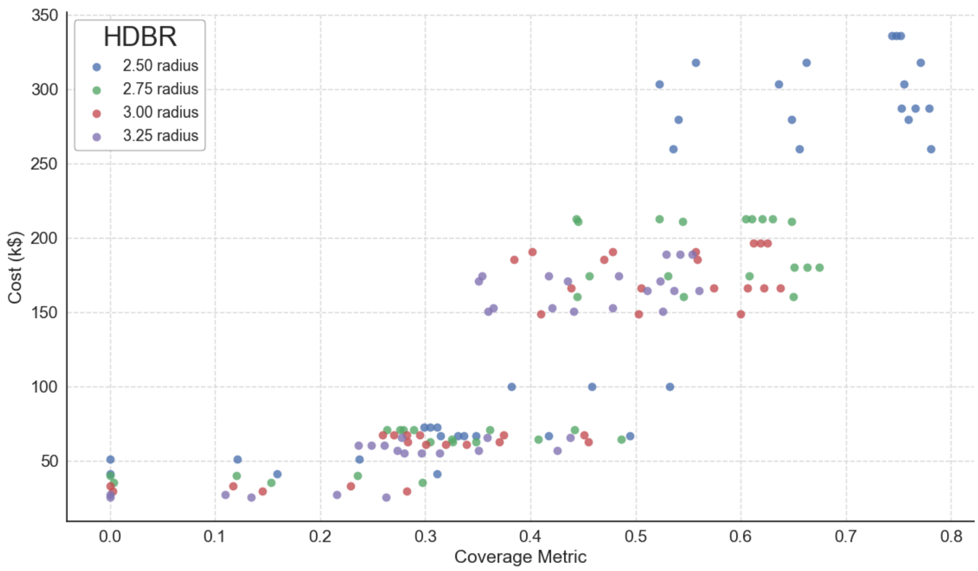


FIGURE 4.4: Impact of the *Horizontal Distance Between Readers* decision on the Architectural Tradespace for the baseline case.

If the horizontal distance between readers is set to 2.50 times the radius, more readers can be fit in the same space, hence increasing the coverage and the cost. As the number of extra readers that can be placed when changing from 2.50 to 2.75 is greater than the extra amount when shifting from 3.00 to 2.75, a gap in the tradespace is created in the high-cost-high-performance region.

We can appreciate how the gap between the architectures with different HDBR options is smaller in the low-cost-low-performance area, as even decreasing the distance between readers is not enough to make a significant change if we just limit the tracking to a small area such as the limited by the check-in counters.

Within all the tradespace, there are gaps between architectures across the performance axis too. Specifically, it is appreciated in the low-cost-low-performance and high-cost-high-performance regions of the tradespace. The reason behind this gap is the interaction between the *Tracking Departing Passengers* and *Tracking Arriving Passengers* decisions, as shown in Figure 4.5.

The figure shows why the gap is only present in the regions mentioned in the previous paragraph. On the one hand, in the low-cost-low-performance area, the readers are placed in the zones where only departing passengers go through. This is why when

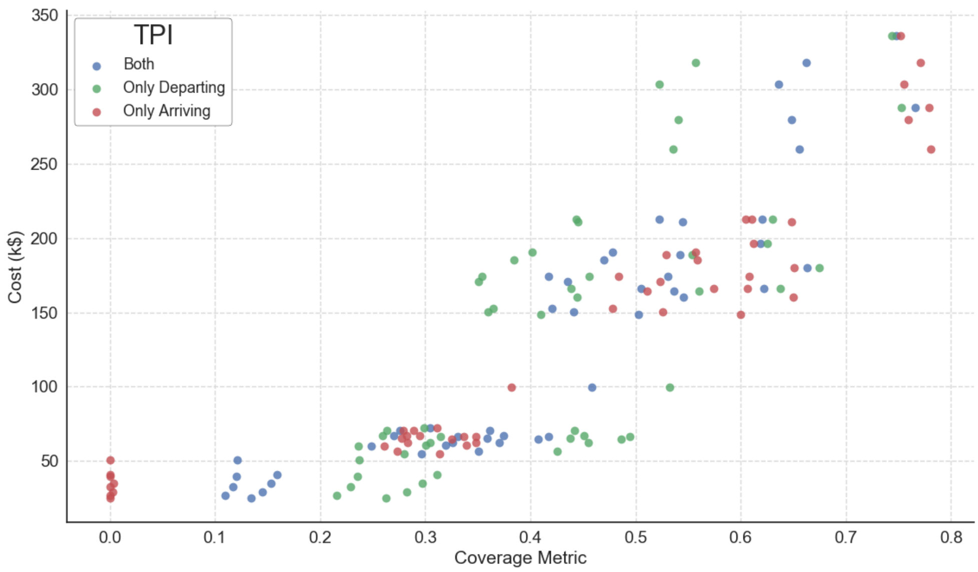


FIGURE 4.5: Impact of the interaction between *Tracking Departing Passengers* and *Tracking Arriving Passengers* decisions on the Architectural tradespace for the baseline case.

tracking only arriving passengers we obtain a set of architectures with zero performance, and when tracking only departing passengers we perform the best.

On the other hand, in the high-cost-high-performance region, due to the fact that arriving passengers' trajectories are shorter and more straight-forward, it is easier to cover more effectively these passengers' trajectories than the departing passengers' ones, which are more irregular and comprise more surface, thus comprising more blind spots too.

Regarding the *Departing Initial Trackin Point* decision, we can appreciate, in Figure 4.6, how starting the tracking in the entrance of the airport has a more robust performance against the TDP and TAP decisions in the high-cost-high-performance region. The reason why most of the architectures in the Pareto Front present the entrance of the terminal as the tracking initial point must not be confused with dominance. In the low-cost-low-performance region, there is no other possible option than starting at the entrance, because in Figure 4.3 we have shown all these architectures finish the tracking at the check-in counters. In the high-cost-high-performance region, where there is presence of the three options for the DITP decision, these architectures are sometimes dominated by the architectures tracking from the security point to the gates. However, from the medium-cost-medium-performance region we can conclude the architectures with DITP = Entrance dominate the architectures with DITP = Check-in Counters.

Finally, we have to consider the impact the *Antennas per Reader* decision has on the tradespace. As showed in Figure 4.7, there is a clear advantage in using 4 antennas instead of 3. In order to achieve the same performance than the 4 antennas architectures,

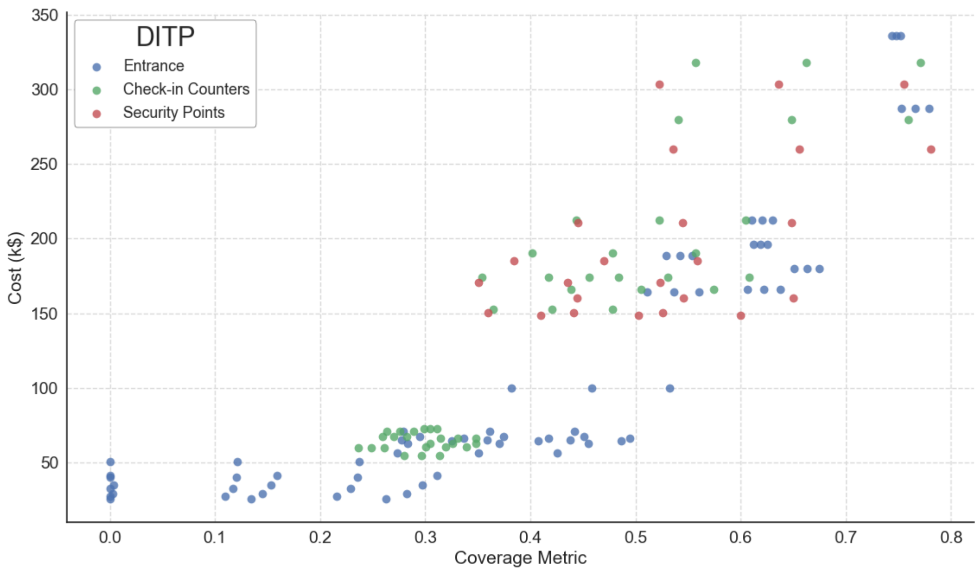


FIGURE 4.6: Impact of the *Departing Initial Tracking Point* decision on the Architectural Tradespace for the baseline case.

the 3 antennas architectures see a substantial increase in its cost. That is confirmed by noticing that all the architectures in the Pareto Front except one present the 4 antennas option for the APR decision.

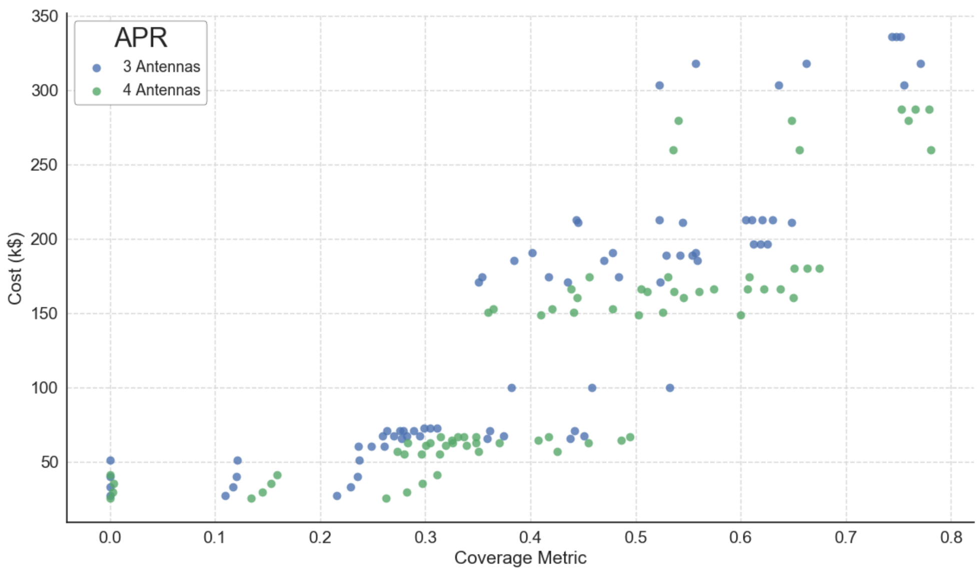


FIGURE 4.7: Impact of the *Antennas Per Reader* decision on the Architectural Tradespace for the baseline case.

At this point, we have observed the different impact each of the architectural decisions has on the tradespace. Some of the decisions have a clear advantageous option, others show their effects in the tradespace as a form of interaction and finally the rest can not be related with dominance and present all of their options in the Pareto Front.

4.2.3 Coverage Metric vs. Tracking Metric

In subsection 3.4.2 we developed two different metrics: The coverage metric, dealing with the architectural decisions directly; and the tracking metric, which although it is not affected by the architecture itself, still is an important metric for the correct functioning of the system.

Therefore, in Figure 4.8 we present the tradespace with the tracking metric in the x-axis for the baseline case. Given the computational expenses of the S-ALOHA simulation, we limit the architectures to the non-dominated architectures in Figure 4.2 and analyze just the 35% of the flight schedule of the baseline case.

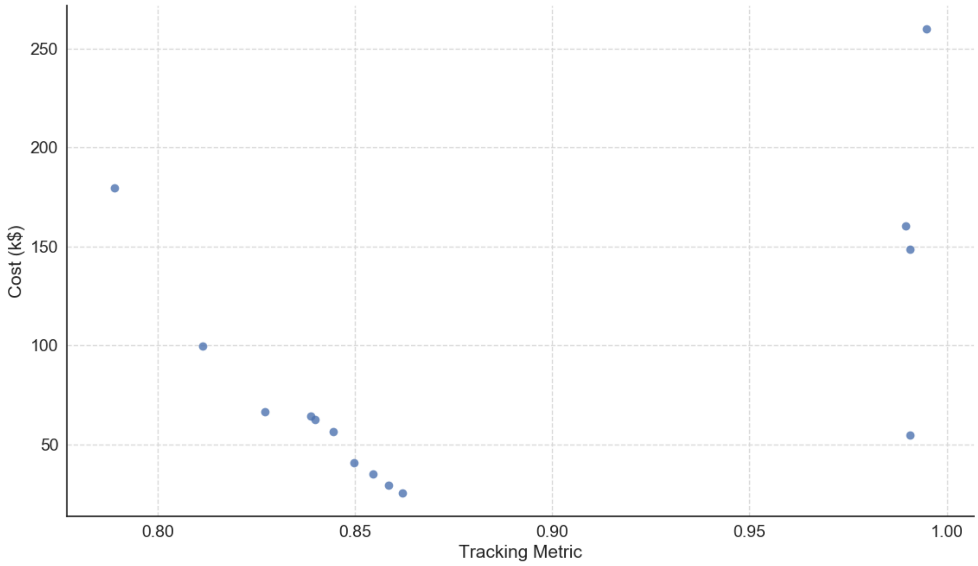


FIGURE 4.8: Architectural Tradespace for the baseline case using the Tracking Metric.

There are two clearly different architecture tendencies regarding the tracking metric. On the one hand, in the rightmost part of the tradespace, four architectures present a tracking performance close to 100%. This four architectures are the ones in which only the arriving passengers are tracked.

On the other hand, all the architectures that just track the departing passengers —there was no architecture tracking both types of passengers in the Pareto Front in Figure 4.2— are in the leftmost part of the tradespace. What is relevant about these architectures is the relationship of inverse proportionality between the cost and performance, where the best architectures in performance are the cheapest ones. The reason behind this non-intuitive result is that a decrease in the cost turns into a decrease in the number of readers, and therefore a decrease in the covered area. Given less area is being covered, there are less passengers covered simultaneously, so the S-ALOHA protocol estimates a smaller number of time slots to provide, thus being able to fit more ALOHA cycles in the 5 seconds that an ID request lasts.

4.3 Sensitivity Analyses

At this point we have thoroughly examined the performance and cost of the 144 architectures of the baseline case and related the impact of each decision on the optimization problem. However, it is important to check how sensitive this results are to the way we have modeled the system and the input assumptions we have made. This is the goal of this section.

4.3.1 Robustness Against the Calendar

One of the inputs of the model is a complete flight schedule extracted from the Chicago O'Hare airport flight schedule webs. In the baseline case, we selected the flight schedule that happened on Thursday, July 27$^{\text{th}}$ 2017. However, it is crucial that the results extracted from the baseline case maintain through different flight schedules, otherwise the system would not be suitable to be operated all year through.

Therefore, we use the flight schedule from Sunday, August 20$^{\text{th}}$ as input for the system. That day, a total of 1,091 flights were scheduled, divided in 537 arrivals and 554 departures. The new architectural tradespace is shown in Figure 4.9.

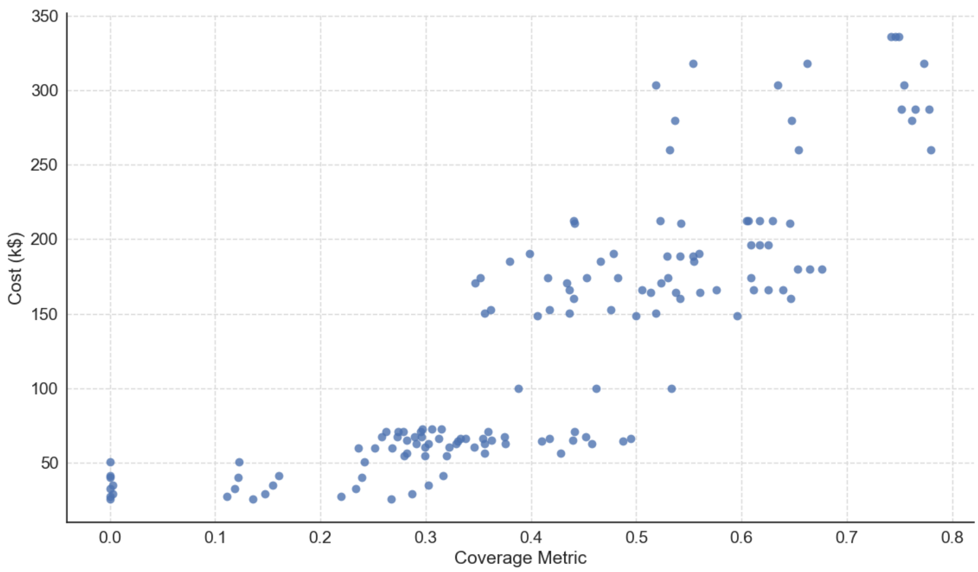


FIGURE 4.9: Architectural Tradespace for the weekend case.

Comparing the new tradespace with the tradespace in Figure 4.1 one could tell no difference between them. To ease that task, in Figure 4.10 we show the percentage of change in the coverage metric for the different architectures.

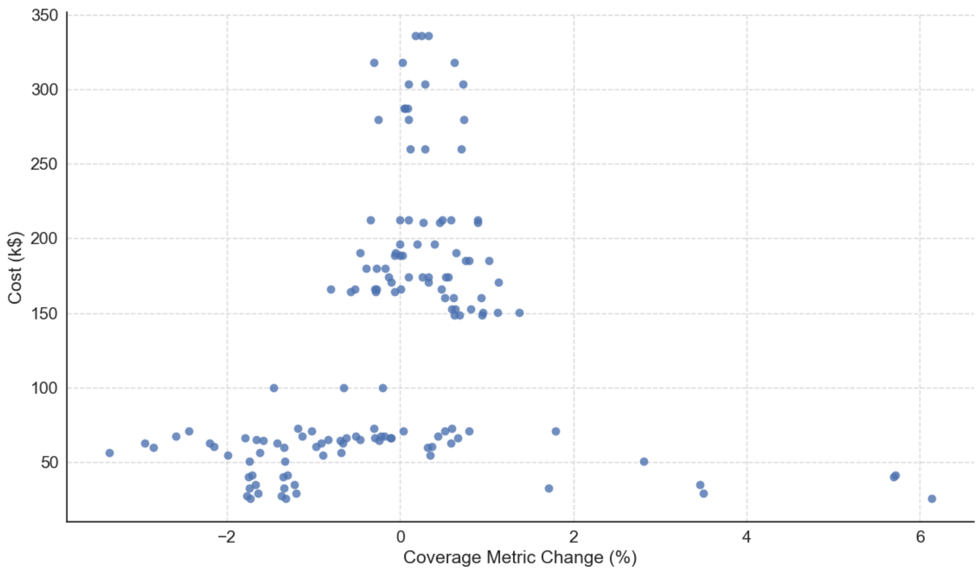


FIGURE 4.10: Percentage of change in the coverage metric between the baseline case and the “august” case.

Most of the architectures have not suffered changes of more than a 4% in the metric, specially the high-cost architectures are more robust in that sense. The more readers in the terminal, the more coverage offered and therefore the probability of being insensitive to different trajectories increases. In the low-cost area, given there are less readers, this changes provoke a greater impact.

After seeing these results, we can say the system seems to be robust to the changes in the flight schedule, guaranteeing the operability of the system during all year through.

4.3.2 Changing the Reader Placement Heuristic

Throughout the development of the model we have introduced algorithms that serve to all architectures but are not affected by the architectural decisions. One of these algorithms is the reader placing heuristic, explained in subsection 3.3.3. To perform another sensitivity analysis we change some parameters of this heuristic in order to optimize for the cost of the architectures —we want to place less readers without changing the performance—. The resulting new tradespace is shown in Figure 4.11.

We can appreciate how the cost of the most expensive architectures has decreased in more than 50 thousand dollars, while maintaining the metrics in coverage. It appears to

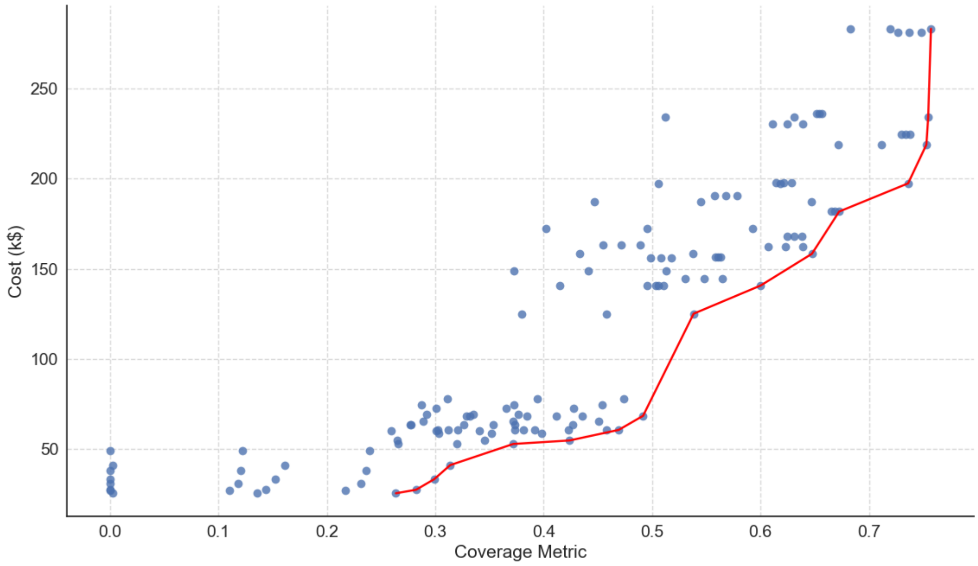


FIGURE 4.11: Architectural Tradespace when changing the reader placement heuristic.

be an asymptote around the 75% of coverage that was not appreciable in the baseline case's Pareto Front. Now, the new Pareto Front shows 16 non-dominated architectures (see Table F.2). For clarity, in Figure 4.12 we present the percentages of the change in cost and performance for each architecture.

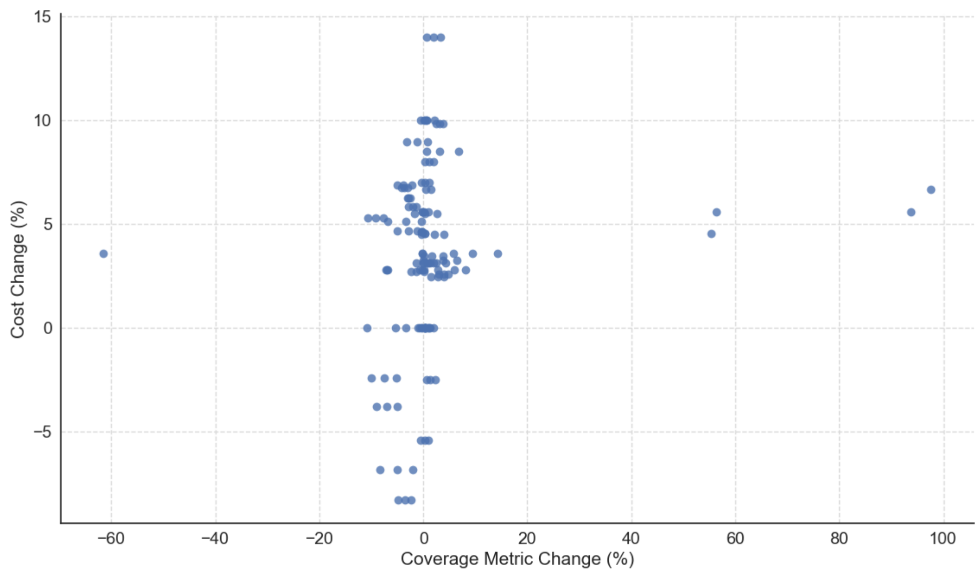


FIGURE 4.12: Percentage of change in the cost and coverage metrics between the baseline case and the "new heuristic" case.

This time, the change in the cost and performance the new heuristic introduces is more notable. We can see how most of the architectures experience performance changes within the 0%-15% range and cost changes in the 0%-10% range. Again, there are a few architectures —outliers— that show changes much larger than the rest, specially in the coverage metric.

4.3.3 Extending the Distance Between Readers

Finally, we realize we have limited the *Horizontal Distance Between Readers* decision to four options, but given it is the only decision defined by non-bounded numbers we decide to extend the amount of options to see how the tradespace changes. So, alongside the changed heuristic from the previous analysis, we introduce 3.50, 4.00, 4.50, 5.00 and 6.00 times the radius as extra options for the HDBR decision. This way, the number of possible options is 9 and the amount of feasible architectures adds up to 324. This new tradespace is shown in Figure 4.13. As a reminder, we can not decrease that distance further than 2.50 times the radius, otherwise we would experience channel interference that would affect the tracking performance of the system.

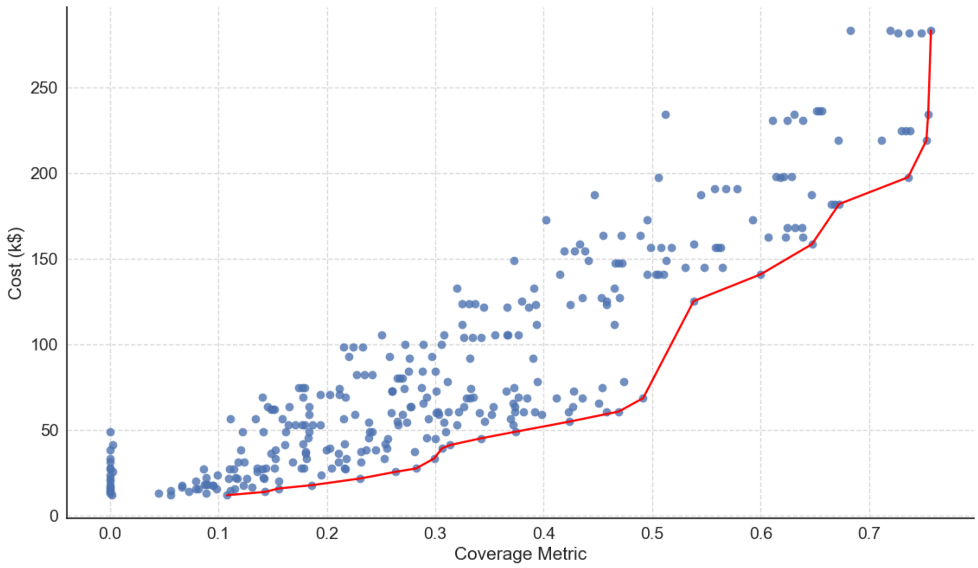


FIGURE 4.13: Architectural Tradespace for the extended *Horizontal Distance Between Readers* decision.

The new Pareto Front is formed by 23 different architectures (see Table F.3), reducing the amount of architectures in the Pareto Front from 9.7% to 7.1%. We can also appreciate how the gaps that have been present during all the previous analyses have now been filled with the extra architectures. In Figure 4.14 we can observe how the architectures with different options now merge and the gaps are no longer present.

Then, in Figure 4.15 we show how all the options for the HDBR decision distribute across the tradespace. While the high-cost-high-performance region is still dominated by the "2.50 radius" architectures, the medium-cost-medium-performance region has

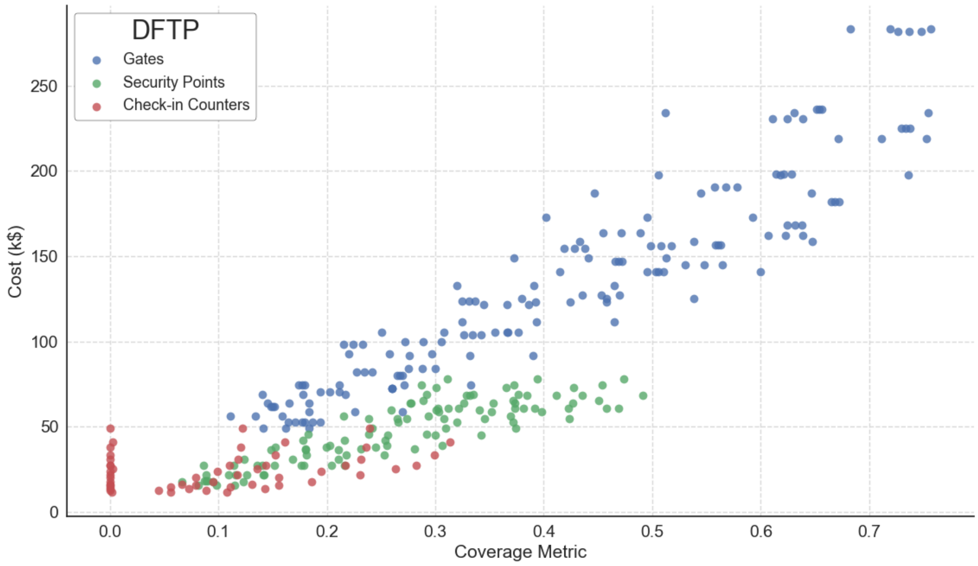


FIGURE 4.14: Impact of the *Departing Final Tracking Point* decision on the Architectural Tradespace for the extended *Horizontal Distance Between Readers* decision.

been filled with the new architectures, presenting a gradient that proportionally relates the performance and the cost. At the region of 70 thousand dollars in cost we can appreciate architectures with “2.50 radius” too. These architectures, as in the baseline case, are the ones that finish the tracking anywhere but the gates.

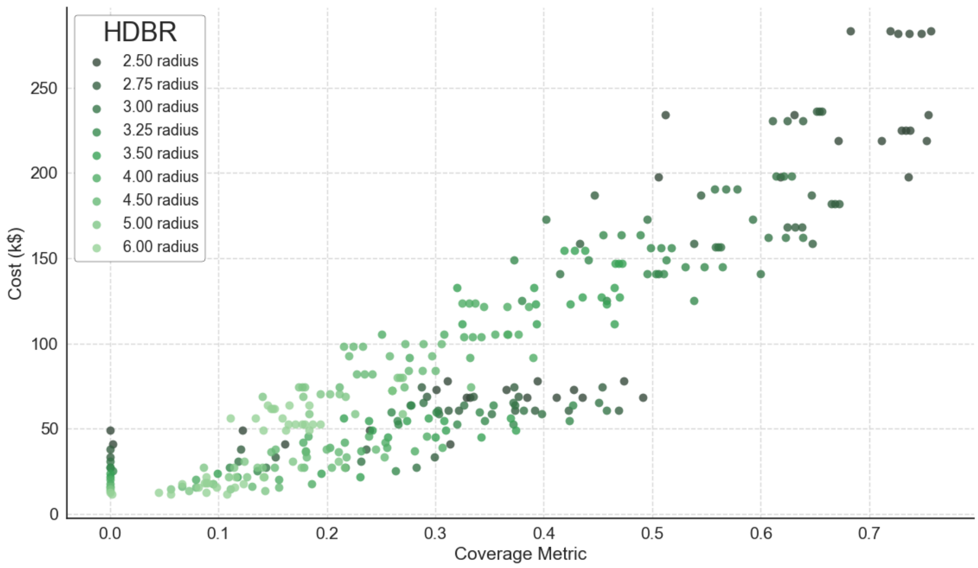


FIGURE 4.15: Impact on the Architectural Tradespace of the extended *Horizontal Distance Between Readers* decision.

Finally, we present Figure 4.16 as the tradespace showing, for each option of the extended HDBR decision, the architecture that accounts for the highest coverage metric and its cost, in the form of the number of readers needed for that architecture. This figure shows a linear tendency, specially in the low-cost region, that serves to estimate the number of necessary readers to achieve a specific amount of coverage.

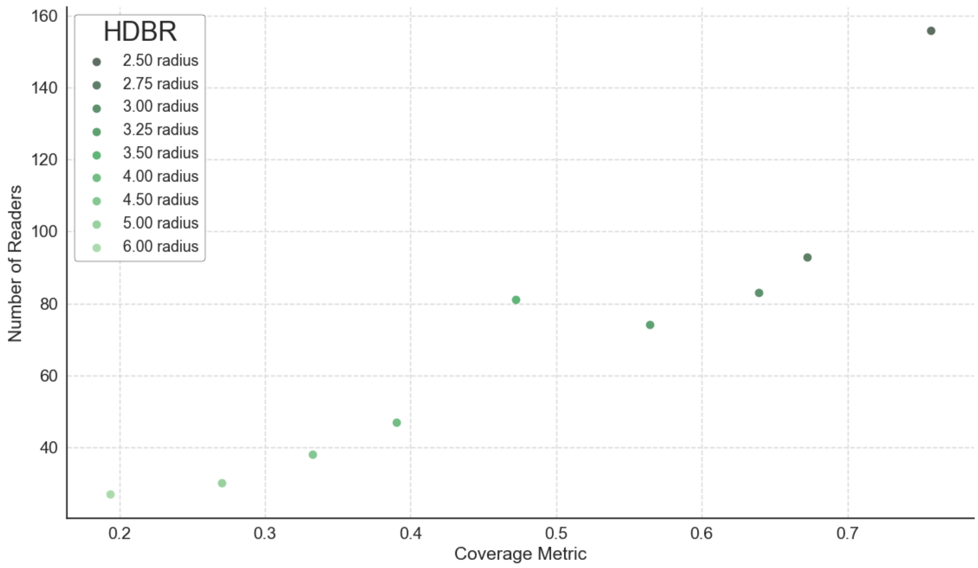


FIGURE 4.16: Maximum Coverage Metric against the Number of Readers for each of the levels of the extended *Horizontal Distance Between Readers* decision.

4.4 Recommendations

In the previous sections we have analyzed how each decision affects the tradespace and how sensitive the system is to the changes in the inputs or the parameters of the model. All this information needs to be translated to effective recommendations for the final implementation and operation of the system in a real terminal.

During the discussion of the impact of each of the decisions, we have been able to observe how the effect of each decision on the tradespace was different. While some of the decisions present a dominating option, others are better understood as an interaction, or their best option depends on further elements.

Beginning this discussion with the *Antennas per Reader* decision, Figure 4.7 has shown how using 4 antennas on each reader clearly outperforms using 3 antennas. There is a notable increase in change accompanied by a subtle increase in cost.

One of the decisions that has showed its effect on the overall tradespace is the *Departing Final Tracking Point* decision. This decision, far from presenting a clear option, grouped the architectures in the baseline case (Figure 4.3) and created a uniform distribution in the extended HDBR case (Figure 4.14). At this point a further analysis is necessary. The goal of this system is to track the passengers inside an airport in order to create passenger density measurements to use for time and security purposes. Given that the vast majority of passengers spend most of their time in the area between the security point and the gates —in fact, the arriving passengers just walk through that area—, we opt for recommending to track until the gates and therefore set the DFTP decision to the “Gates” option.

Then, the *Tracking Departing Passengers* and *Tracking Arriving Passengers* decisions have showed their effect in the form of interaction. Given we have set the DFTP decision to “Gates” thus focusing on the high-cost-high-performance region, we should opt for just tracking the arriving passengers, given this option has had better results for that region, as showed in Figure 4.5. However, in Figure 4.6 we have seen how choosing to start the tracking from the entrance of the terminal reduces the gaps in performance that the interaction of TDP and TAP creates. Therefore, if the placement of the readers is done across all the airport —from the entrance to the gates—, we not only capture the greatest part of an airport’s activity, but also reduce the sensitivity of the performance against who we want to track.

This way, the airport managers are not limited to select who they want to track before the deployment of the infrastructure. In fact, they can choose who they want to track, or give RFID tags to, at any time.

Finally, the impact of the *Horizontal Distance Between Readers* decision on the tradespace (Figure 4.4) shows no preference for any of the considered options. We can appreciate how in the high-cost-high-performance region no option clearly dominates, therefore suggesting a selection process based on further constraints such as budget or spacing limitations.

In Table 4.1 we show the summary of the recommendations presented in this section.

Architectural Decision	**Recommendation**
Tracking Departing Passengers (TDP)	As preferred by the airport manager at each time
Tracking Arriving Passengers (TAP)	As preferred by the airport manager at each time
Departing Initial Tracking Point (DITP)	Tracking passengers from the entrance
Departing Final Tracking Point (DFTP)	Tracking passengers until the gates
Antennas per Reader (APR)	Using 4 antennas on each reader
Horizontal Distance Between Readers (HDBR)	No dominating option, choose based on further criteria

Table 4.1: Recommendations for each of the architectural decisions modeled.

The reader may remember we concluded the architectural decisions section with more decisions that the ones showed in the table. Specifically, we are talking about the *Tracking Connecting Passengers*, *Tag Technology* and *Number of ALOHA Cycles* decisions. Given that the system is robust against who it tracks, it is fair to assume the tracking of the connecting passengers will not be a problem. Regarding the technology incorporated in the tag, in subsection 3.4.1 we showed why the *Higgs-EC* IC dominated the rest of the ICs. Finally, also in subsection 3.4.1 we determined the derisory impact the number of ALOHA cycles has on the cost metric and chose to set a time limit instead of a cycle quantity limit. The complete set of recommendations is showed in Table 4.2.

Architectural Decision	**Recommendation**
Tracking Departing Passengers (TDP)	As preferred by the airport manager at each time
Tracking Arriving Passengers (TAP)	As preferred by the airport manager at each time
Tracking Connecting Passengers (TCP)	As preferred by the airport manager at each time
Departing Initial Tracking Point (DITP)	Tracking passengers from the entrance
Departing Final Tracking Point (DFTP)	Tracking passengers until the gates
Antennas per Reader (APR)	Using 4 antennas on each reader
Horizontal Distance Between Readers (HDBR)	No dominating option, choose based on further criteria
Tag Technology (TT)	The *Higgs-EC* IC dominates the rest of available ICs
Number of ALOHA Cylces (NOAC)	It is better to set a time limit, as there is no important impact on the cost

TABLE 4.2: Recommendations for all the architectural decisions considered in the beginning.

Chapter 5

Conclusions

With the presentation of the tradespace results in the previous chapter we reach the scope of this Thesis. The purpose of this chapter is to summarize the main points of this Thesis, in section 5.1; list the main findings from this research, in section 5.2; and finally discuss the future research on the topic as well as the extension of this Thesis, in section 5.3.

5.1 Summary

This Thesis has explored the main research procedures of System Architecture applied to a new concept that is *Internet of Aerospace Things*. We have initiated the work with a brief description of IoAT and its goals. Then, after listing some IoAT envisioned projects based on the aeronautics industry, we have decided to pursue a system to help, by means of connectivity, in the passengers mobility and security inside an airport terminal.

In chapter 2, we have started with the discussion on the solution-neutral function for this system and decided it to be "Reducing Passengers Traverse/Transfer Time". Then, we have selected Tracking using RFID as the concept for the system and followed with an introduction to the RFID technology, highlighting the most important parts for the interest of this research, being the physical layer, the data link layer and the regulations on RFID.

After the theoretical introduction on the technology, we have come up with the set of formal and functional, architectural decisions that define each of the possible architectures and encoded them in the form of a morphological matrix. Also, we have showed the logical constraints that reduce the number of feasible architectures.

In chapter 3 we have focused on the model created to evaluate each of the feasible architectures. The model presents a common part that all architectures share, being the airport terminal and the passengers that walk across it. Second, we have showed how the RFID communications have been modeled and how each architecture differs with the others based on its option for each decision. To that end, we have presented the link budget, the coverage, the reader placement and the communication protocol submodels. Finally, based on the information exposed so far in the Thesis, we have selected the different performance and cost metrics, that define the location of each architecture on the tradespace.

Before introducing the tradespace, we have initiated chapter 4 wit the description of the baseline case and the inputs of the model. Then, we have computed the metrics for each architecture in the baseline case and displayed the architectural tradespace. After that, we have focused on each of the decisions separately in order to analyze its impact on the tradespace. Furthermore, the best architectures have been analyzed in terms of their ability to correctly track the passengers.

Finally, we have concluded the results chapter with a few sensitivity analyses focusing on the change of the inputs of the model, the change of the parameters of the model and the extension of one of the decisions. After thoroughly analyzing the different tradespaces and the impacts of each decision we have concluded the research with the recommendations for the system.

5.2 Main Findings

The principal task of this Thesis has been to come up with a suitable architecture for a specific IoAT system. We have showed how analyzing this problem from the System Architecture point of view has led to the optimization of the architecture in terms of cost and performance, as we have been able to make a recommendation for eight out of the nine decisions we posed at the beginning of this Thesis. We have obtained that 9.7% of the architectures are non-dominated for the baseline case while this quantity decreases to 7.1% for the extended HDBR case.

The main performance metric during the analyses has been the coverage metric, and it has been shown that the best architectures achieve coverages of 78% in the baseline case or 76% in the sensitivity analyses. These values prove it is possible to cover an important part of the terminal. If we combined this research with actual data of the passenger's movements inside the Chicago airport, we would be able to place the same amount of readers in the most dense spots and therefore achieve even more coverage.

We have also considered a second performance metric, the tracking metric, with which we have shown that the architectures in the Pareto Front allow the tracking function with errors in the range of 14%-20% for the architectures that just track departing passengers and errors below the 2% for the architectures that just track the arriving passengers. This tests have been performed assuming a 5 seconds ALOHA period, arising the possibility of increasing the tracking success rate by increasing the mentioned period.

Finally, based on the tradespace analyses performed in the Results chapter, we have obtained the following conclusions:

1. Using 4 antennas on each reader dominates the use of only 3 antennas.
2. The *Departing Final Tracking Point* decision sets the cost ranges for the different groups of architectures and affects to the effect of other decisions such as the types of passengers to track.
3. There is an interaction between the decisions of tracking the arriving and the departing passengers, respectively, that makes the architectures that just track arriving passengers the best architectures in the high-cost-high-performance region whereas tracking only the departing passengers is the best option for the low-cost-low-performance region.
4. The *Departing Initial Tracking Point* decision can provide robustness to the system against who it decides to track if we focus on the high-cost-high-performance region.
5. No option dominates in the *Horizontal Distance Between Readers* decision and further criteria need to be applied. This includes considering the budget limits or the feasibility of placing the readers where we have exactly placed them in the model.
6. The system is robust against the change in the flight schedule input —specially the high-cost architectures—, although extreme cases need to be tested.
7. Changing some parameters of the model such as the reader placement heuristic can lead to the change in the maximum cost and/or performance values.
8. We can appreciate a linear tendency in the low-cost region that relates the total coverage with the number of readers placed. In the high-cost region, there might be a performance threshold caused by the 50 channel limitation.

5.3 Future Work

One of the main characteristics of this Thesis has been its interdisciplinarity, the involvement of different disciplines, methodologies and tools to obtain insighful results. Given this presence of multiple science and engineering fields, this research leaves the door open for future projects to deepen the studies carried out in the Thesis or to broaden its scope. This future work includes:

- By means of an increase in computational resources, some models such as the reader placement model or the passengers' walking model shall be refined and enhanced.
- A validation of the walking models and ALOHA models needs to be done via real scenario experiments [79].
- The model shall be improved by the introduction of a multipath fading submodel [4].
- The repercussion of using a 2D airport model of a 3D space needs to be further analyzed.
- Further considerations regarding RFID such as the logistics of the distribution of tags or avoiding tags to be occluded or poorly oriented need to be discussed before the system's installation [106].
- Using more than one antenna at each reader leads to the exact positioning of the passengers, which can be exploited to increase the information on passenger density [57].
- With the information the system provides on the airport density, an app has to be created in order to transfer this information to the passengers.
- Further applications such as a machine learning-based program to detect anomalous trajectories or personalized advertisement in the airport may be based on the information from our system.

Appendix A

Principles of System Architecture

Principle of Emergence: "As the entities of a system are brought together, their interaction will cause function, behavior, performance, and other intrinsic properties to emerge" [25].

Principle of Holism: "Every system operates as a art of one large system or several larger systems, and each is itself composed of smaller systems" [25].

Principle of Focus: "The number of identifiable issues that will influence a system at any point is beyond one's ability to understand. One must identify the most critical and consequential issues, and focus on them" [25].

Principle of Dualism: "All built systems inherently and simultaneously exist in the physical domain and the informational domain" [25].

Principle of Benefit Delivery: "Good architectures deliver benefit, first and foremost, built on the primary externally delivered function of the system by focusing on the emergence of functions, and their delivery across the system boundary at an interface" [25].

Principle of Value and Architecture: "Value is benefit at cost. Architecture is function enabled by form. There is a very close relationship between these two statements, because benefit is delivered by function, and forms is associated with cost" [25].

Principle of Solution-Neutral Function: "Poor system specifications frequently contain clues about an intended solution, function, or form, and these clues may lead the architect to a narrower set of potential options. Use solution-neutral functions where possible, and use the hierarchy of solution-neutral statements to scope how broad an exploration of the problem is to be undertaken" [25].

Principle of the Role of the Architect: "The role of the architect is to resolve ambiguity, focus creativity, and simplify complexity" [25].

Principle of Ambiguity: "The early phase of a system design is characterized by great ambiguity. The architect must resolve this ambiguity to produce (and continuously update) goals for the architect's team" [25].

Principle of the Stress of Modern Practice: "Modern product development process, with concurrency, distributed teams, and supplier engagement, places even more emphasis on having a good architecture" [25].

Principle of Architectural Decisions: "Separate architectural decisions from other decisions, and take the time to carefully decide them up front, because they will be very expensive to change later on" [25].

Principle of Reuse of Legacy Elements: "Understand the legacy system and its emergent properties thoroughly, and include the necessary elements in the new architecture" [25].

Principle of Product Evolution: "Systems will evolve or lose competitive advantage. When architecting, define the interfaces as the more stable parts of the system so that the elements can evolve" [25].

Principle of the Beginning: "The list of stakeholders (internal or external to the enterprise) that are included in the early stages of product definition will have an outsized impact on the architecture" [25].

Principle of Balance: "Many factors influence and act on the conception, design, implementation, and operation of a system. One must find a balance among the factor that satisfies the most important stakeholders" [25].

Principle of the System Problem Statement: "The statement of the problem defines the high-level goal and establishes the boundaries of the system. Challenge and refine the statement until you are satisfied that it is correct" [25].

Principle of Ambiguity and Goals: "The architect must resolve this ambiguity to produce (and continuously update) a small set of representative, complete and consistent, challenging yet attainable, and humanly solvable goals" [25].

Principle of Creativity: "Creativity in architecture is the process of resolving tensions in the pursuit of good architecture" [25].

Principle of Apparent Complexity: "Create decomposition, abstraction, and hierarchy to keep the apparent complexity within the range of human understanding" [25].

Principle of Essential Complexity: "Functionality drives essential complexity. Describe the required functionality carefully, and then choose a concept that produces low complexity" [25].

Principle of the 2nd Law: "The actual complexity of the system always exceeds the essential complexity. Try to keep the actual complexity close to the essential complexity" [25].

Principle of Decomposition: "Decomposition is an active choice made by the architect. The decomposition affects how performance is measured, how the organization should be set up, and the potential for supplier value capture" [25].

Principle of "2 Down, 1 Up": "The goodness of a decomposition at Level 1 cannot be evaluated until the next level down has been populated and the relationships identified (Level 2)" [25].

Principle of Elegance: "Elegance is appreciated internally by the architect when a system has a concept with low essential complexity and a decomposition that aligns many of the planes of decomposition simultaneously" [25].

Principle of Robustness of Architectures: "Good architectures can respond to change by being robust (capable of dealing with variations in the environment) or by being adaptable (able to adapt to changes in the environment)" [25].

Principle of Coupling and Organization of Architectural Decisions: "The sequence of architectural decisions can be chosen by considering the sensitivity of the metrics to the decisions and the degree of connectivity of decisions" [25].

Appendix B

System Architecture Definitions

Concept: "Concept is a product or system vision, idea, notion, or mental image that maps function to form. It is a scheme for the system and how it works. It embodies a sense of how the system will function and an abstraction of the system form. It is a simplification of the system architecture that allows for high-level reasoning. Concept is not a product/system attribute but a notional mapping between two attributes: form and function" [25].

Context: "What surrounds the system. It is the entities that are "just on the outside of the system" but are relevant to it" [25].

Form: "The physical or informational embodiment of a system that exists or has the potential for stable, unconditional existence, for some period of time, and is instrumental in the execution of function. Form exists prior to the execution of function" [25].

Function: "The activity, operation, or transformation that causes or contributes to performance. In designed systems, function is the actions for which as system exist, which ultimately lead to the delivery of value. Function is executed by form, which is instrumental in function. Function emerges from functional interaction between entities" [25].

Object: "That which has the potential for stable, unconditional existence for some period of time" [25].

Operand: "An object. Therefore it has the potential for stable, unconditional existence for some period of time. Operands are objects that need not exist prior to the execution of function and are in some way acted upon by the function. Operands may be created, modified, or consumed by the process part of function" [25].

Process: "A pattern of transformation undergone by an object. Processes generally involve creation of, destruction of, or a change in an operand" [25].

Structure: "Relationships between objects of form that have the potential for stable, unconditional existence for some duration of time and may be instrumental in the execution of functional interactions. Also called formal relationships" [25].

System Architecture: "The embodiment of concept, the allocation of physical/informational function to the elements of form, and the definition of relationships among the elements and with the surrounding context" [25].

Appendix C

Object-Process Methodology: Building Blocks and Links

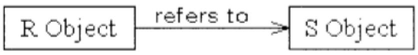
The Building Blocks

	Visual Representation	Textual Form	Definition	Description
Entities	Object	Nouns; capitalized first letter in every word; if ending with "ing", "Object" is placed as a suffix	*An **object** is a thing that has the potential of stable, unconditional physical or mental existence.*	Static things. Can be changed only by processes.
	Process(ing)	Nouns in gerund form; capitalized first letter in every word; if not ending with "ing", "Process" is placed as a suffix	*A **process** is a pattern of transformation that an object undergoes.*	Dynamic things. Are recognizable by the changes they cause to objects.
	Object state	Nouns, adjectives or adverbs; non-capitalized	*A **state** is a situation an object can be at.*	States describe objects. They are attributes of objects. Processes can change an object's state.
OPL Conventions	**Non-Reserved Words**	Arial bold by default	*Names of entities.*	Words used by the system architect, unique to the system.
	reserved words	Arial by default; non-capitalized	*Object-Process Language (OPL) words.*	Words or phrases used by OPL, the same in every sentence of a certain type.

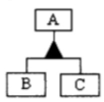
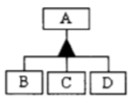
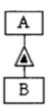
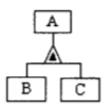
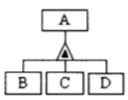
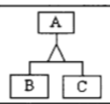
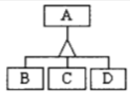
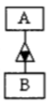
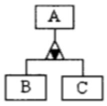
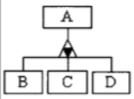
Links: The Mortar

Tagged Structural Links

Generally used between objects, but may also be used between processes.
Cannot be used to link an object to a process.

Link Name	Object Process Diagram (OPD) Symbol	OPL Sentence	Description
Tagged		**R Object refers to S Object.**	Relation from source object to destination object; relation name is entered by architect, and is recorded along link.
(Null)		**R Object** relates to **S Object.**	Relation from source object to destination object with no tag.
Bi-directional Tagged		**R Object precedes S Object. S Object follows R Object.**	Relation between two objects; relation names are entered by architect, and are recorded along link.
(Null) Bi-directional		**R Object** and **S Object** are related.	Relation between two objects with no tag.

A fundamental structural relation can have many descendants.
The different OPL sentences and OPD pictures are listed below.

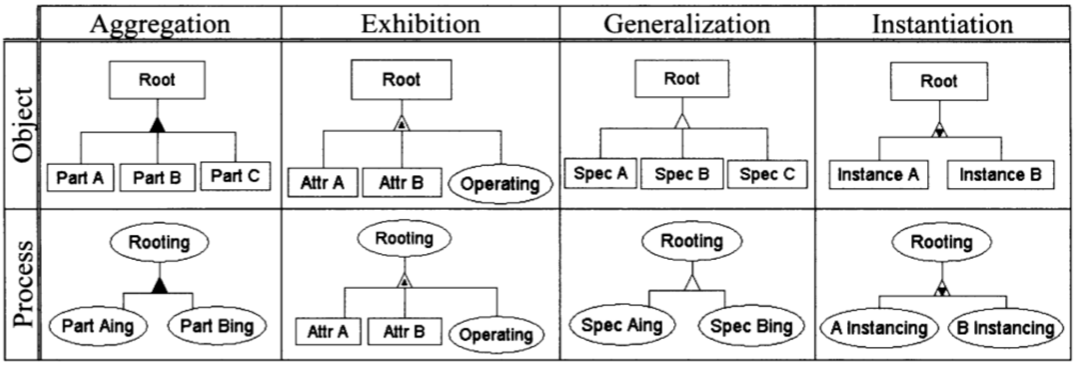
Structural Relation Name and **Shorthand Name**	Number of Descendants						Description
	One		Two		Three or more		
	OPD	OPL	OPD	OPL	OPD	OPL	
Aggregation-Participation		**A** consists of **B.**		**A** consists of **B** and **C.**		**A** consists of **B, C,** and **D.**	B, C and D are parts of the whole A.
Exhibition-Characterization		**A** exhibits **B.**		**A** exhibits **B** and **C.**		**A** exhibits **B, C,** and **D.**	B, C and D are attributes of A. If B is a process, it is an operation of A.
Generalization-Specialization		**B** is an **A.**		**B** and **C** are **A**s.		**B, C,** and **D** are **A**s.	B, C and D are types of A.
Classification-**Instantiation**		**B** is an instance of **A.**		**B** and **C** are instances of **A.**		**B, C,** and **D** are instances of **A.**	B, C and D are unique objects of the class A.

Links: The Mortar (continued)

The Four Fundamental Structural Relations

Shorthand Name	Aggregation	Exhibition	Generalization	Instantiation
Symbol				
Meaning	Relates a whole to its parts	Relates an exhibitor to its attributes	Relates a general thing to its specializations	Relates a class of things to its instances

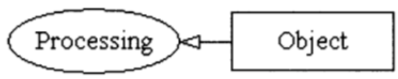
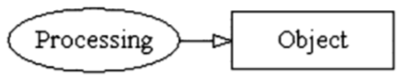
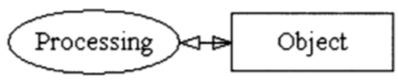
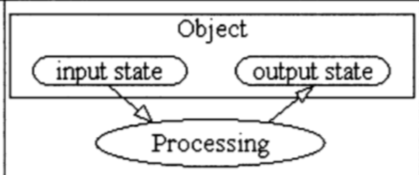
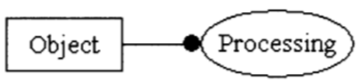
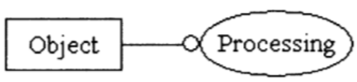
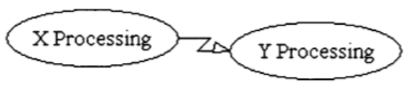
The four fundamental relations are also applicable to processes. Only exhibition can link objects with processes. Instantiation cannot generate a hierarchy while the other three can. Any number of things can be linked to the root.



Links: The Mortar (continued)

Procedural Links

These links are generally used between an object and a process. They cannot be used to link objects together.

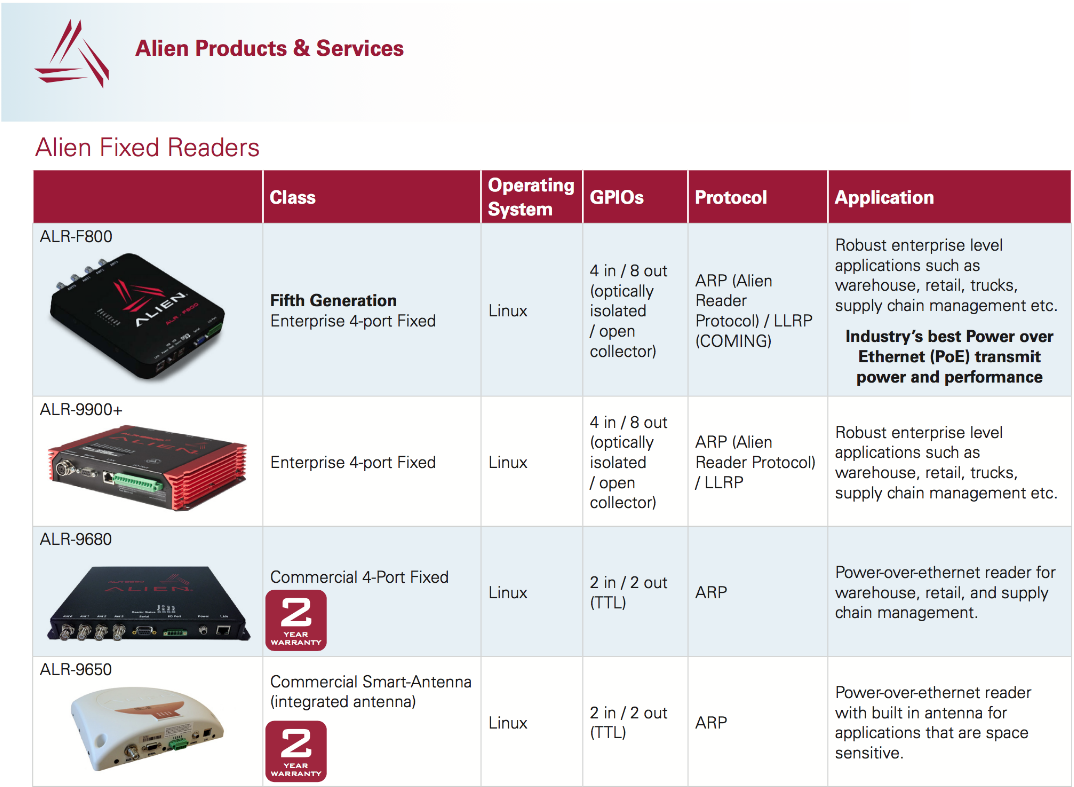
Link Name	OPD Symbol	OPL Sentence	Description
Consumption		**Processing** consumes **Object.**	Process uses object up entirely during its occurrence.
Result		**Processing** yields **Object.**	Process creates an entirely new object during its occurrence.
Effect		**Processing** affects **Object.**	Process changes the state of the object in an unspecified manner.
Input and Output		**Processing** changes **Object** from **input state** to **output state.**	The object is at input state prior to the process occurrence, and at output state as a result of its occurrence.
Agent		**Object** handles **Processing.**	Object is a human that is not changed by the process; process needs the agent object in order to occur.
Instrument		**Processing** requires **Object.**	Object is a non-human that is not changed by the process; process needs the instrument object in order to occur.
Invocation		**X Processing** invokes **Y Processing.**	First process directly starts up a second process, without an inter-mediate object.

Appendix D

RFID Products Datasheets

D.1 RFID Readers

Figure D.1 shows the available RFID readers in the RFID seller's product catalog, alongside a brief feature description and usual applications.


Alien Products & Services

Alien Fixed Readers

	Class	Operating System	GPIOs	Protocol	Application
ALR-F800	**Fifth Generation** Enterprise 4-port Fixed	Linux	4 in / 8 out (optically isolated / open collector)	ARP (Alien Reader Protocol) / LLRP (COMING)	Robust enterprise level applications such as warehouse, retail, trucks, supply chain management etc. **Industry's best Power over Ethernet (PoE) transmit power and performance**
ALR-9900+	Enterprise 4-port Fixed	Linux	4 in / 8 out (optically isolated / open collector)	ARP (Alien Reader Protocol) / LLRP	Robust enterprise level applications such as warehouse, retail, trucks, supply chain management etc.
ALR-9680	Commercial 4-Port Fixed 2 YEAR WARRANTY	Linux	2 in / 2 out (TTL)	ARP	Power-over-ethernet reader for warehouse, retail, and supply chain management.
ALR-9650	Commercial Smart-Antenna (integrated antenna) 2 YEAR WARRANTY	Linux	2 in / 2 out (TTL)	ARP	Power-over-ethernet reader with built in antenna for applications that are space sensitive.

Figure D.1: Different RFID reader models offered.

D.2 RFID Antennas

Figure D.2 presents the different antennas offered while comparing them in frequency range, gain, IP rating, size and cabling features.

Alien Products & Services

Alien Reader Antenna

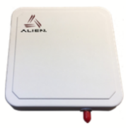
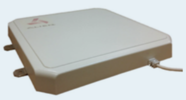
	Frequency	Gain	IP Rating	Size	Cable/Connector
ALR-A1001 	865-867MHz 902-928MHz	8.5dBic	IP67	9.84″ x 9.84″ x 0.55″	Inset SMA male connector
ALR-A0501 	865-868MHz 902-928MHz	6dBic	IP67	5.04″ x 5.04″ x 0.79″	Inset SMA male connector
ALR-8697 	865 - 928 MHz	8.5dBic	IP67	10.16″ x 10.16″ x 1.42″	Inset RTNC male connector
ALR-8698 		Nominally 10.5dBic 11dBic FCC / 10dBic ETSI			
ALR-8696 	865 - 960 MHz	8.5dBic	IP54	10.2″ x 10.2″ x 1.32″	20ft LMR-195 cable with RTNC Female at the end. Connected to antenna via a pigtail.

Figure D.2: Different antenna models offered.

D.3 RFID Tags

In Figure D.3 a descriptive figure of the available RFID tags in the RFID seller's catalog is presented. All tags are introduced and properly classified depending on the tag family (e.g., *Squiggle*), the model of integrated circuit (e.g., *Higgs-4*) and a qualitative performance measure of different RFID applications.

	Squiggle Family					Specialized Retail					High-Dielectric / Automotive				Form-factor			
	Squiggle® (SQ)	Short (SH)	Squiglette (ST)	Squig (SG)	Doc Tag (DT)	Glint (GL)	Pearl (PE)	HiScan (HS)	GT-Tag (GT)	Spider-360 (SP)	BAT-Tag (BAT)	Wonder-Dog (WD)	G-Tag (G)	Bio-Tag	SlimLine (SL)	2x2	Square (1x1)	SIT (SIT)
Higgs™-EC Model #	ALN-9840	N/A	ALN-9830	N/A	N/A	N/A	N/A	ALN-9820	ALN-9828	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Higgs™-4 Model #	ALN-9740	ALN-9762	ALN-9730 ALN-9730-	ALN-9710	ALN-9741	ALN-9715	ALN-9716	ALN-9720	ALN-9728	ALN-9726	ALN-9770	ALN-9768	N/A	ALN-9714	ALN-9745	N/A	N/A	ALN-9713
Higgs™-3 Model #	ALN-9640	ALN-9662	ALN-9630	ALN-9610	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	ALN-9654	N/A	N/A	ALN-9634	ALN-9629	ALN-9613
Retail	BEST	BEST	BEST	BEST	BETTER	BEST	BEST	BEST	BEST	BEST	BETTER	BETTER	BETTER	GOOD	BETTER	BETTER	GOOD	BETTER
Mobile Asset Tracking*	BEST	BEST	BEST	BEST	BETTER	GOOD	GOOD	BETTER	GOOD	BETTER	BETTER	BEST	BETTER	GOOD	BETTER	BETTER	BETTER	N/A
Automotive	BETTER	BETTER	BETTER	BETTER	BETTER	GOOD	GOOD	BETTER	GOOD	GOOD	BEST	BEST	BEST	BETTER	BEST	GOOD	GOOD	N/A
Pharma / Healthcare	BEST	BEST	BETTER	BEST	BEST	BEST	BEST	BEST	BETTER	BETTER	BETTER	BETTER	BETTER	BEST	BEST	BETTER	BEST	BEST
Ultra compact	N/A	N/A	GOOD	BETTER	N/A	BETTER	BETTER	BETTER	N/A	N/A	N/A	N/A	N/A	BEST	BEST	N/A	GOOD	BEST
Wood/ Plastic/ Glass	BETTER	BETTER	BETTER	BETTER	BETTER	GOOD	GOOD	BEST	BETTER	BETTER	BEST	BEST	BEST	BEST	BETTER	BETTER	BETTER	BETTER
Files/ Paper	BEST	BETTER	BETTER	GOOD	BEST	N/A	N/A	BETTER	GOOD	BETTER	N/A	BETTER	N/A	N/A	BETTER	GOOD	GOOD	N/A
General Purpose	BEST	BEST	BEST	BEST	BETTER	BETTER	BETTER	BETTER	GOOD	BEST	GOOD	BEST	BETTER	N/A	BETTER	BEST	BETTER	N/A
Ref Design Available?	Yes	Yes	Yes	Yes	No	Yes	No	No	No	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes

BEST BETTER GOOD N/A

FIGURE D.3: Different RFID tag models offered.

D.4 Tag Integrated Circuits

Figure D.4 shows the main characteristics that define each of the current RFID tag's integrated circuit in the market.

Higgs™ IC Family

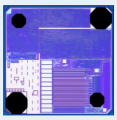
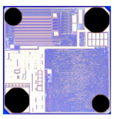
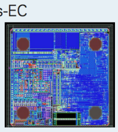
	Memory	EPC bits	UTID	Sensitivity	Serialization	Applications
Higgs 3 	800 bits	96-480 bits	64 bits	Very good (-20dBm)	Yes - Chip & Non-chip serialization	General purpose, apparel, Automatic Vehicle Identification (AVI), automotive, baggage, books, cases, documents, ID cards, item level tracking, labels, loyalty cards, pharmaceutical, palettes, retail etc
Higgs 4 	512 bits	128 bits	64 bits	Excellent (-20.5dBm)		
Higgs-EC 	512 bits	128 bits	80 bits	Best-In-Class (-22.5dBm)		
	Error Correcting Memory					

FIGURE D.4: Differences between the main integrated circuits available for tags.

Appendix E

Five Basic Airport Passenger Building Configurations

In this appendix we present and discuss the five basic airport passenger building configurations from [6], providing real examples and the simulation models we considered at the beginning of this project.

Designing an airport requires the consideration of the different stakeholders and their respective interests. In the end, the designer faces a tradeoff between the need of passengers of not walking excessive distances and the interest of airlines in having enough space to operate their airplanes comfortably. Existing airports prove five basic configurations for the passenger building exist. These are: Finger Piers, Satellites, Midfield Concourses, Linear Buildings and Transporters.

E.1 Finger Piers

"*Finger Pier configurations* are simply relatively narrow extensions to a central passenger facility. In plan view as seen from the air, they resemble fingers attached to the palm of a hand. Their obvious form places aircraft gates on both sides of the building extending away from the central core. This arrangement has the advantage of placing some aircraft gates close to the central facility, and thus more conveniently for the passengers than the gates at the end of the finger pier." [6]

Examples of airports with finger pier configurations are New York La Guardia, Chicago O'Hare, San Francisco International, London Heathrow or Paris Orly. In Figure E.1 we show the finger terminal we designed for the first stages of the project, insipired by Barcelona El Prat airport.

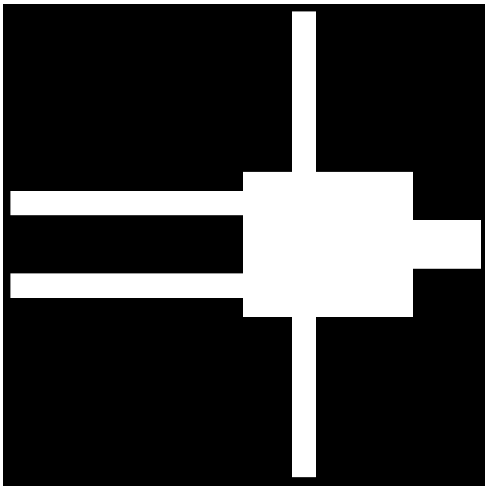


FIGURE E.1: Finger Pier configuration, inspired by Barcelona El Prat Airport.

E.2 Satellites

"*Satellites* are the logical extension of T-shaped finger piers. They eliminate the gates along the fingers and concentrate gates at the end. Generally, the connection between the satellite and the central check-in area is aboveground. In some designs, designers place the finger underground and it becomes invisible. The satellite is sometimes connected to the central part of the passenger building by a people mover, sometimes not." [6]

Airports like Paris Charles de Gaulle, Seattle Tacoma, Tokyo Narita or Milan Malpensa present satellites configurations with different arrangements. Inspired by New York Newark Airport, we developed the mock terminal showed in Figure E.2 during the first stages of the project.

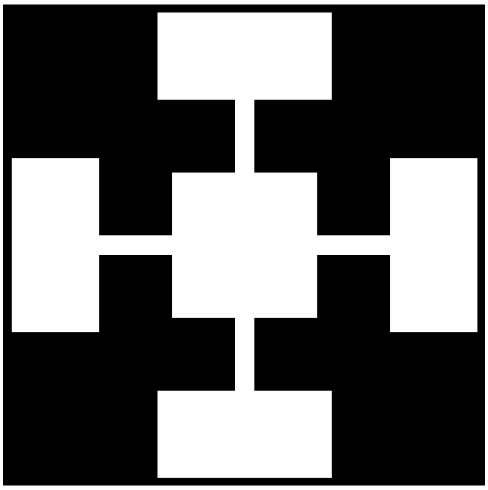


FIGURE E.2: Satellites configuration, inspired by New York Newark Airport.

E.3 Midfield Concourses

"*Midfield concourses* are major independent passenger buildings, often located far from the passenger building, which passengers access from the groundside. They may easily have around 50 gates and be about a kilometer long. Midfield concourses come in two basic shapes: linear and X-shaped. Linear concourses are simply long buildings with aircraft positions on both sides. X-shaped midfield concourses feature intersecting fingers that give them the X-shape. Normally, the crosspieces are oriented at about 45 and 135^{o} with respect to parallel runways that flank the midfield concourses." [6]

We can find examples of midfield concourses in airports such as Pittsburgh International, Atlanta Hartsfield-Jackson, Hong Kong Chek Lap Kok or London Stansted. We based our first design of a midfield concourse terminal on Denver International Airport, as shown in Figure E.3.

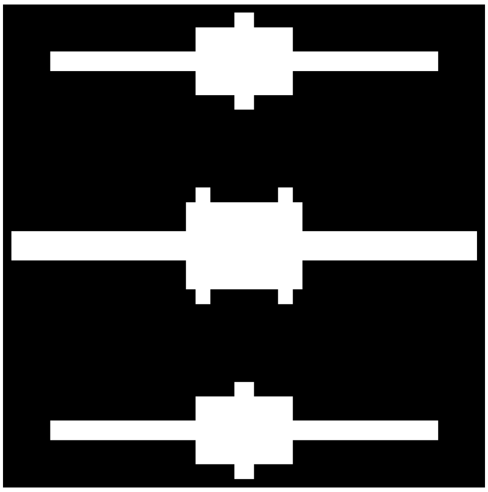


FIGURE E.3: Midfield configuration, based on Denver International Airport.

E.4 Linear Buildings

"A *linear building* is a long, relatively thin structure with one side devoted to aircraft and the other faced by roads and parking lots. Designers came up with the concept of linear buildings in response to the great walking distances associated with finger piers. They originally called it the "gate arrival" concept. The idea was that people could drive or be driven right up to their departure gate, park their cars if necessary, and get to their flight by walking through a narrow building". [6]

Airports such as Dallas Fort Worth or Kansas City Airport built gate arrival or linear buildings in the past. The mock terminal used as a linear building example was based on Munich-Franz Josef Strauss International Airport, shown in Figure E.4.

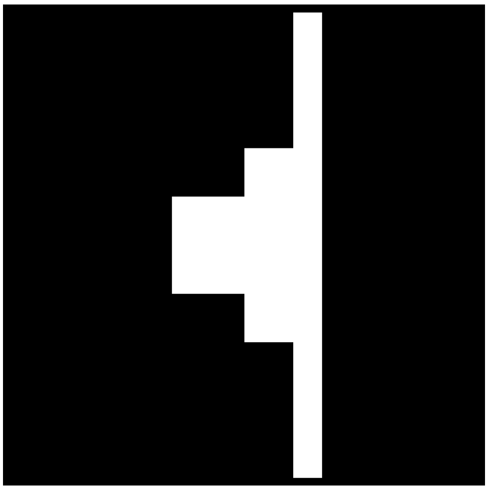


FIGURE E.4: Linear configuration, based on Munich International Airport.

E.5 Transporters

"*Transporters* are the broad category of rubber-tired vehicles that move passengers between passenger buildings and aircraft. Most simply, these are specially designed buses with low platforms and wide aisles for easy access for passengers with bags. These airport buses require passengers to walk up and down the stairs between the airport apron and the aircraft door." [6]

In this type of configurations lie airports such as Berlin's or Zürich's, which use this vehicles to displace passengers through the airport. These terminals distinguish for its simplicity, as observed in the mock terminal used in Figure E.5, based on Washington-Dulles Airport.

FIGURE E.5: Transporters configuration, based on Washington-Dulles Airport.

Appendix F

Non-Dominated Architectures

F.1 Baseline Case

Number	Metrics		Architectural Decisions					
	Performance	Cost (k$)	TDP	TAP	DITP	DFTP	APR	HDBR
1	0.781	260.0	No	Yes	SP	G	4	2.50
2	0.675	179.9	Yes	No	E	G	4	2.75
3	0.650	160.3	No	Yes	SP	G	4	2.75
4	0.600	148.6	No	Yes	SP	G	4	3.00
5	0.532	99.9	Yes	No	E	SP	3	2.50
6	0.494	66.5	Yes	No	E	SP	4	2.50
7	0.486	64.5	Yes	No	E	SP	4	2.75
8	0.455	62.6	Yes	No	E	SP	4	3.00
9	0.425	56.7	Yes	No	E	SP	4	3.25
10	0.314	54.7	No	Yes	CC	SP	4	3.25
11	0.311	41.1	Yes	No	E	CC	4	2.50
12	0.297	35.2	Yes	No	E	CC	4	2.75
13	0.282	29.3	Yes	No	E	CC	4	3.00
14	0.263	25.4	Yes	No	E	CC	4	3.25

TABLE F.1: List of non-dominated architectures for the baseline case.

E = Entrance

CC = Check-in Counters

SP = Security Point

G = Gates

F.2 Change in the Reader Placement Heuristic

Number	Metrics		Architectural Decisions					
	Performance	Cost (k$)	TDP	TAP	DITP	DFTP	APR	HDBR
1	0.756	283.3	No	Yes	CC	G	3	2.50
2	0.755	234.2	No	Yes	SP	G	3	2.50
3	0.754	219.0	No	Yes	CC	G	4	2.50
4	0.736	197.5	No	Yes	SP	G	4	2.50
5	0.673	181.8	Yes	No	E	G	4	2.75
6	0.647	158.4	No	Yes	SP	G	4	2.75
7	0.597	140.8	No	Yes	SP	G	4	3.00
8	0.537	125.1	No	Yes	SP	G	4	3.25
9	0.492	68.4	Yes	No	E	SP	4	2.50
10	0.470	60.6	Yes	No	E	SP	4	2.75
11	0.426	54.7	Yes	No	E	SP	4	3.25
12	0.374	52.8	Yes	No	CC	SP	4	3.25
13	0.317	41.1	Yes	No	E	CC	4	2.50
14	0.303	33.2	Yes	No	E	CC	4	2.75
15	0.286	27.4	Yes	No	E	CC	4	3.00
16	0.267	25.4	Yes	No	E	CC	4	3.25

TABLE F.2: List of non-dominated architectures for heuristic change case.

E = Entrance

CC = Check-in Counters

SP = Security Point

G = Gates

F.3 Distance Between Readers Decision Extended

Number	Metrics		Architectural Decisions					
	Performance	Cost (k$)	TDP	TAP	DITP	DFTP	APR	HDBR
1	0.756	283.3	No	Yes	CC	G	3	2.50
2	0.755	234.2	No	Yes	SP	G	3	2.50
3	0.754	219.0	No	Yes	CC	G	4	2.50
4	0.736	197.5	No	Yes	SP	G	4	2.50
5	0.673	181.8	Yes	No	E	G	4	2.75
6	0.647	158.4	No	Yes	SP	G	4	2.75
7	0.597	140.8	No	Yes	SP	G	4	3.00
8	0.537	125.1	No	Yes	SP	G	4	3.25
9	0.492	68.4	Yes	No	E	SP	4	2.50
10	0.470	60.6	Yes	No	E	SP	4	2.75
11	0.426	54.7	Yes	No	E	SP	4	3.25
12	0.376	48.9	Yes	No	E	SP	4	3.50
13	0.343	45.0	Yes	No	CC	SP	4	3.50
14	0.317	41.1	Yes	No	E	CC	4	2.50
15	0.308	39.1	Yes	No	E	SP	4	4.00
16	0.303	33.2	Yes	No	E	CC	4	2.75
17	0.286	27.4	Yes	No	E	CC	4	3.00
18	0.267	25.4	Yes	No	E	CC	4	3.25
19	0.234	21.5	Yes	No	E	CC	4	3.50
20	0.188	17.6	Yes	No	E	CC	4	4.00
21	0.157	15.6	Yes	No	E	CC	4	4.50
22	0.145	13.7	Yes	No	E	CC	4	5.00
23	0.109	11.7	Yes	No	E	CC	4	6.00

TABLE F.3: List of non-dominated architectures for the extended HDBR decision case.

E = Entrance

CC = Check-in Counters

SP = Security Point

G = Gates

Bibliography

[1] Yifeng Han, Qiang Li, and Hao Min. System modeling and simulation of RFID. In *Auto-ID Labs Research Workshop*, page 24, 2004.

[2] Vipul Chawla and Dong Ha. An overview of passive RFID. *IEEE Communications Magazine*, 45(9):11–17, sep 2007.

[3] Klaus Finkenzeller. *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication.* John Wiley & Sons, 2010.

[4] Daniel Arnitz. *Tag Localization In Passive UHF RFID.* PhD thesis, Graz University of Technology, 2011.

[5] Google Maps. https://www.google.com/maps/.

[6] Richard de Neufville. *Airport Systems Planning.* Palgrave Macmillan UK, London, 1976.

[7] Kin Seong Leong, Mun Leng Ng, and P.H. Cole. Positioning Analysis of Multiple Antennas in a Dense RFID Reader Environment. In *International Symposium on Applications and the Internet Workshops (SAINTW'06)*, pages 56–59. IEEE.

[8] Harald Vogt. Efficient Object Identification with Passive RFID Tags. pages 98–113. 2002.

[9] Chip Walter. Kryder's Law. *Scientific American*, 293(2):32–33, 2005.

[10] Gordon E. Moore. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86(1):82–85, 1998.

[11] Jakob Nielsen. Nielsen's law of internet bandwidth. *Online at http://www. useit. com/alertbox/980405. html*, 1998.

[12] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things: A survey. *Computer Networks*, 54(15):2787–2805, oct 2010.

[13] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[14] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. Internet of things: Vision, applications and research challenges, 2012.

[15] Debasis Bandyopadhyay and Jaydip Sen. Internet of things: Applications and challenges in technology and standardization. In *Wireless Personal Communications*, volume 58, pages 49–69, 2011.

[16] Cary R Spitzer and Cary Spitzer. *Digital Avionics Handbook*. CRC press, 2000.

[17] Ashton Morrow. Delta introduces innovative baggage tracking process. http://news.delta.com/delta-introduces-innovative-baggage-tracking-process-0, 2016.

[18] Boeing's Airplane Health Management. http://www.boeing.com/assets/pdf/commercial/aviationserv

[19] Tomas Kellner. Follow The Digital Thread: How GE's Off To See Profit In The Age Of Cheap Oil. http://www.ge.com/reports/follow-the-digital-thread-how-ges-off-to-see-profit-in-the-age-of-cheap-oil/.

[20] Aieron and Iridium NEXT Partnership. https://aireon.com/resources/iridium-next/.

[21] Michael Thomas. Flight attendant tech personalizes the skies. http://news.delta.com/flight-attendant-tech-personalizes-skies, 2015.

[22] WSI Total Turbulence. https://business.weather.com/products/total-turbulence.

[23] Mark W Maier. *The art of systems architecting*. CRC press, 2009.

[24] Edward Crawley, O de Weck, Steven Eppinger, Christopher Magee, Joel Moses, Warren Seering, Joel Schindall, David Wallace, Daniel Whitney, O Weck de, Steven Eppinger, Christopher Magee, Joel Moses, Warren Seering, Joel Schindall, David Wallace, Daniel Whitney, and Olivier De Weck. The influence of architecture in engineering systems. *MIT Engineering Systems Symposium*, pages 1–30, 2004.

[25] Edward Crawley, Bruce Cameron, and Daniel Selva. *System architecture: Strategy and product development for complex systems*. Prentice Hall Press, 2015.

[26] Dov Dori. *Object-Process Methodology*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[27] Dov Dori. *Model-Based Systems Engineering with OPM and SysML.* Springer New York, New York, NY, 2016.

[28] Marc Sanchez Net, Iñigo del Portillo, Bruce Cameron, and Edward F Crawley. Architecting Information Security Services for Federated Satellite Systems. *Journal of Aerospace Information Systems*, pages 1–12, 2016.

[29] Marc Sanchez Net, Inigo del Portillo, Bruce G Cameron, Edward F Crawley, and Daniel Selva. Integrated tradespace analysis of space network architectures. *Journal of Aerospace Information Systems*, 12(8):564–578, 2015.

[30] Markus Guerster. Architectural Options and Optimization of Suborbital Space Tourism Vehicles. Master's thesis, Technical University of Munich, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Munich, Germany, 2017.

[31] D J Power. Decision support systems: concepts and resources for managers. *Studies in Informatics and Control*, 11(4):349–350, 2002.

[32] Fritz Zwicky, Fritz Zwicky, and Others. Discovery, invention, research through the morphological approach. 1969.

[33] Donald V Steward. The design structure system: A method for managing the design of complex systems. *IEEE transactions on Engineering Management*, (3):71–74, 1981.

[34] Inigo del Portillo, Marc Sanchez, Bruce G Cameron, and Edward F Crawley. Optimal Location of Optical Ground Stations to Serve LEO Spacecraft. In *2017 IEEE Aerospace Conference*, 2017.

[35] H. Zimmermann. OSI Reference Model–The ISO Model of Architecture for Open Systems Interconnection. *IEEE Transactions on Communications*, 28(4):425–432, apr 1980.

[36] Leonard Kleinrock. *Queueing systems, volume 2: Computer applications*, volume 66. wiley New York, 1976.

[37] Anke Fransen. Connector, Brussels Airport's most striking innovation of 2015! http://www.amcham.be/publications/amcham-connect/2015/february/brussels-airport-connector, 2015.

[38] Ben Mutzabaugh. Austin airport's new indoor/outdoor South Terminal is now open. https://www.usatoday.com/story/travel/flights/todayinthesky/2017/04/14/austin-airports-new-indooroutdoor-south-terminal-is-now-open/100472072/, 2017.

[39] National Audit Office. E-borders and successors programmes. https://www.nao.org.uk/wp-content/uploads/2015/12/E-borders-and-successor-programmes.pdf, 2015.

[40] SITA Press Release. SITA app takes Miami International Airport to new heights. https://www.sita.aero/pressroom/news-releases/getting-personal-sita-app-takes-miami-international-airport-to-new-heights, 2016.

[41] Boston Logan's interactive map. http://maps.massport.com.

[42] Tiancheng Zhang, Yifang Yin, Dejun Yue, Xirui Wang, and Ge Yu. Research and implementation of an rfid simulation system supporting trajectory analysis. *Journal of Software*, 9(1):162–168, 2014.

[43] Ricardo O. Mitchell, Hammad Rashid, Fakir Dawood, and Ali Alkhalidi. Hajj crowd management and navigation system: People tracking and location based services via integrated mobile and RFID systems. *International Conference on Computer Applications Technology, ICCAT 2013*, 2013.

[44] Nobuaki Ishihara, Huijing Zhao, and Ryosuke Shibasaki. Tracking passenger movement with ground-based laser scanner. In *Paper presented at the 22nd Asian Conference on Remote Sensing*, volume 5, page 9, 2001.

[45] Yasunori Yakiyama, Niwat Thepvilojanapong, Masayuki Iwai, Oru Mihirogi, Kazunori Umeda, and Yoshito Tobe. Observing Real-world Attention by a Laser Scanner. 4:515–528, 2009.

[46] P Falcone, F Colone, a Macera, and P Lombardo. Localization and tracking of moving targets with WiFi-based passive radar. *Radar Conference (RADAR), 2012 IEEE*, pages 705–709, 2012.

[47] Brussels Airport aims to cut queues with passenger tracking sensors. http://www.futuretravelexperience.com/2015/02/brussels-airport-to-cut-queues-with-passenger-tracking-sensors/, 2015.

[48] Sunkyu Woo, Seongsu Jeong, Esmond Mok, Linyuan Xia, Changsu Choi, Muwook Pyeon, and Joon Heo. Application of WiFi-based indoor positioning system for labor tracking at construction sites: A case study in Guangzhou MTR. *Automation in Construction*, 20(1):3–13, 2011.

[49] Saad M. Khan and Mubarak Shah. Tracking multiple occluding people by localizing on multiple scene planes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):505–519, 2009.

[50] Martijn Liem and Dariu M. Gavrila. Multi-person tracking with overlapping cameras in complex, dynamic environments. *Procedings of the British Machine Vision Conference 2009*, (January):87.1–87.10, 2009.

[51] Ziyan Wu and Richard J. Radke. Real-time airport security checkpoint surveillance using a camera network. In *CVPR 2011 WORKSHOPS*, pages 25–32. IEEE, jun 2011.

[52] G. Roussos. Enabling RFID in Retail. *Computer*, 39(3):25–30, mar 2006.

[53] Rebecca Angeles. Rfid Technologies: Supply-Chain Applications and Implementation Issues. *Information Systems Management*, 22(1):51–65, dec 2005.

[54] Zheng Mingxiu, Fu Chunchang, and Yang Minggen. The Application used RFID in Third Party Logistics*. *Physics Procedia*, 25:2045–2049, 2012.

[55] G Brookner. RFID device tracking and information gathering, 2005.

[56] George E. Vastianos, Dimitris M. Kyriazanos, Olga E. Segou, Stelios A. Mitilineos, and Stelios C. A. Thomopoulos. Indoor localization using passive RFID. page 80501R, may 2011.

[57] C. Alippi, D. Cogliati, and G. Vanini. A statistical approach to localize passive RFIDs. In *2006 IEEE International Symposium on Circuits and Systems*, page 4. IEEE.

[58] Kin Seong Leong, Mun Leng Ng, A.R. Grasso, and P.H. Cole. Synchronization of RFID Readers for Dense RFID Reader Environments. In *International Symposium on Applications and the Internet Workshops (SAINTW'06)*, pages 48–51. IEEE.

[59] George E. Vastianos, Dimitris M. Kyriazanos, Vassilios I. Kountouriotis, and Stelios C. A. Thomopoulos. An RFID-based luggage and passenger tracking system for airport security control applications. page 90911A, jun 2014.

[60] A Kovach. Method for tracking and processing passengers and their transported articles, 2005.

[61] T. McCoy. RFID for airport security and efficiency. In *IEE Seminar on Signal Processing Solutions for Homeland Security*, volume 2005, pages 8–8. IEE, 2005.

[62] Gemma Simpson. New RFID tech would track airport passengers. https://www.cnet.com/news/new-rfid-tech-would-track-airport-passengers/, 2005.

[63] Rhea Wessel. Airport Monitoring System Combines RFID With Video. http://www.rfidjournal.com/articles/view?2658, 2006.

[64] David Millward. Airports to track passengers with radio ID tags. http://www.telegraph.co.uk/news/uknews/1548100/Airports-to-track-passengers-with-radio-ID-tags.html, 2007.

[65] R. Want. An Introduction to RFID Technology. *IEEE Pervasive Computing*, 5(1):25–33, jan 2006.

[66] J.D. Day and H. Zimmermann. The OSI reference model. *Proceedings of the IEEE*, 71(12):1334–1340, 1983.

[67] J.L. Martinez Flores, Satya Sai Srikant, B. Sareen, and A. Vagga. Performance of RFID tags in near and far field. In *2005 IEEE International Conference on Personal Wireless Communications, 2005. ICPWC 2005.*, pages 353–357. IEEE.

[68] Roy Want. RFID Explained: A Primer on Radio Frequency Identification Technologies. *Synthesis Lectures on Mobile and Pervasive Computing*, 1(1):1–94, jan 2006.

[69] John G Proakis, Masoud Salehi, Ning Zhou, and Xiaofeng Li. *Communication systems engineering*, volume 2. Prentice Hall New Jersey, 1994.

[70] P.V. Nikitin and K.V.S. Rao. Performance limitations of passive UHF RFID systems. *2006 IEEE Antennas and Propagation Society International Symposium*, pages 1011–1014, 2006.

[71] Abdelmoula Bekkali, Horacio Sanson, and Mitsuji Matsumoto. RFID indoor positioning based on probabilistic RFID map and Kalman Filtering. In *3rd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2007*, 2007.

[72] Andreas F Molisch. *Wireless communications*, volume 34. John Wiley & Sons, 2012.

[73] R. Bultitude. Measurement, characterization and modeling of indoor 800/900 MHz radio channels for digital communications. *IEEE Communications Magazine*, 25(6):5–12, jun 1987.

[74] T.S. Rappaport and C.D. McGillem. UHF fading in factories. *IEEE Journal on Selected Areas in Communications*, 7(1):40–48, jan 1989.

[75] M. Zorzi and L. Tomba. A comparison of CDMA, TDMA and slotted Aloha multiple access schemes in cellular mobile radio systems. In *5th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Wireless Networks - Catching the Mobile Future.*, volume 3, pages 776–780. IOS Press.

[76] Michael Buettner and David Wetherall. An Empirical Study of UHF RFID Performance. *MobiCom*, pages 223–234, 2008.

[77] Kin Seong Leong, Mun Leng Ng, and P.H. Cole. Operational considerations in simulation and deployment of RFID systems. In *2006 17th International Zurich Symposium on Electromagnetic Compatibility*, pages 521–524, 2006.

[78] Alien Technology. http://www.alientechnology.com.

[79] Christian Floerkemeier and Ravikanth Pappu. Evaluation of RFIDSim - A physical and logical layer RFID simulation engine. In *2008 IEEE International Conference on RFID (Frequency Identification), IEEE RFID 2008*, pages 350–356, 2008.

[80] Christian Floerkemeier and Sanjay Sarma. RFIDSim - A physical and logical layer simulation engine for passive RFID. *IEEE Transactions on Automation Science and Engineering*, 6(1):33–43, 2009.

[81] Daeyoung Kim, M.A. Ingram, and W.W. Smith. Measurements of small-scale fading and path loss for long range RF tags. *IEEE Transactions on Antennas and Propagation*, 51(8):1740–1749, aug 2003.

[82] R. V. Levine and A. Norenzayan. The Pace of Life in 31 Countries. *Journal of Cross-Cultural Psychology*, 30(2):178–205, 1999.

[83] Betty J. Mohler, William B. Thompson, Sarah H. Creem-Regehr, Herbert L. Pick, and William H. Warren. Visual flow influences gait transition speed and preferred walking speed. *Experimental Brain Research*, 181(2):221–228, 2007.

[84] H. Rahmalan, M.S. Nixon, and J.N. Carter. On crowd density estimation for surveillance. In *IET Conference on Crime and Security*, volume 2006, pages 540–545. IEE, 2006.

[85] N Doshi and R Moriyama. Application of simulation models in airport facility design. In *Simulation Conference, 2002. Proceedings of the Winter*, volume 2, pages 1725–1730 vol.2, 2002.

[86] Ju Yanbing, Wang Aihua, and Che Haiying. Simulation and optimization for the airport passenger flow. In *2007 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2007*, pages 6599–6602, 2007.

[87] Alexander Verbraeck and Edwin Valentin. Simulation building blocks for airport terminal modeling. In *Simulation Conference, 2002. Proceedings of the Winter*, volume 2, pages 1199–1206. IEEE, 2002.

[88] Chicago O'Hare International Airport. http://www.airport-ohare.com.

[89] Werner W K Hoeger, Laura Bond, Lynda Ransdell, Jane M. Shimon, and Sunitha Merugu. One-mile step count at walking and running speeds. *ACSM's Health and Fitness Journal*, 12(1):14–19, 2008.

[90] Michel R Gatersleben and Simon W der Weij. Analysis and simulation of passenger flows in an airport terminal. In *Simulation Conference Proceedings, 1999 Winter*, volume 2, pages 1226–1231. IEEE, 1999.

[91] Ka Keung Lee, Maolin Yu, and Yangsheng Xu. Modeling of human walking trajectories for surveillance. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 2, pages 1554–1559. IEEE.

[92] Ronan Boulic, Nadia Magnenat Thalmann, and Daniel Thalmann. A global human walking model with real-time kinematic personification. *The Visual Computer*, 6(6):344–358, nov 1990.

[93] Laurent Bezault, Ronan Boulic, Nadia Magnenat-Thalmann, and Daniel Thalmann. An interactive tool for the design of human free-walking trajectories. *Computer Animation92*, pages 87–104, 1992.

[94] Peter Hart, Nils Nilsson, and Bertram Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[95] W. Zeng and R. L. Church. Finding shortest paths on real road networks: the case for A*. *International Journal of Geographical Information Science*, 23(4):531–543, apr 2009.

[96] Keng Hsien Chen, Jui Hsien Yen, Jenn Hwan Tarng, and Chang Fa Yang. A low-profile dipole RFID tag antenna with lateral omni-directional radiation properties for cold-chain logistics applications. *IEEE Antennas and Propagation Society, AP-S International Symposium (Digest)*, pages 1522–1523, 2014.

[97] Salehin Kibria, Mohammad Tariqul Islam, and Baharuddin Yatim. New compact dual-band circularly polarized universal RFID reader antenna using ramped convergence particle swarm optimization. *IEEE Transactions on Antennas and Propagation*, 62(5):2795–2801, 2014.

[98] Mark S Daskin. *Network and discrete location: models, algorithms, and applications*. John Wiley & Sons, 2011.

[99] Qiang Guan, Yu Liu, Yiping Yang, and Wenshneg Yu. Genetic Approach for Network Planning in the RFID Systems. *Sixth International Conference on Intelligent Systems Design and Applications (ISDA '06)*, 2:567–572, 2006.

[100] Melanie Mitchell. *An introduction to genetic algorithms.* MIT press, 1998.

[101] Larry Raisanen and Roger M. Whitaker. Comparison and evaluation of multiple objective genetic algorithms for the antenna placement problem. *Mobile Networks and Applications*, 10(1):79–88, 2005.

[102] Su-Ryun Lee, Sung-Don Joo, and Chae-Woo Lee. An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification. In *The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 166–172. IEEE, 2005.

[103] Andrew Tanenbaum. *Computer Networks.* Prentice Hall Professional Technical Reference, 4th edition, 2002.

[104] Chicago Airport Departures. http://www.airport-ohare.com/departures.php.

[105] Chicago Airport Arrivals. http://www.airport-ohare.com/arrivals.php.

[106] Robert H. Clarke, Diana Twede, Jeffrey R. Tazelaar, and Kenneth K. Boyer. Radio frequency identification (RFID) performance: the effect of tag orientation and package contents. *Packaging Technology and Science*, 19(1):45–54, jan 2006.