Document name:	**Use Case description**
Document version:	**1.0**
Task code:	**T5.4**
Deliverable code:	**D5.4.1**
WP Leader (organisation):	**PetalsLink**
Deliverable Leader (organisation):	**PetalsLink**
Authors (organisations):	**Christophe GATTI (THA)**
Date of first version:	**14/05/12**

Document history

Changes	Author / Entity	Code of version
Creation of the document	Christophe GATTI	0.1
Outline added	Christophe GATTI	0.2
‘Draft review” version	Christophe GATTI	1.0

Abstract

The purpose of this deliverable is to describe the “Air traffic flow management” use-case of the SocEDA project and clarify the individual actions that take place among different actors that it is composed of. People and services interact with each other throughout the entire sequence of steps that compose the use-case.

As such, a fined-grained design of choreographies based on BPMN2.0 is introduced. It is supplemented by SOAML models for the design of service contracts and participants pertaining to these choreographies. Moreover, these SOAML models specify requirements, Quality of Service and interface agreements, as well as the sequencing of the exchanges among participants.

This document also introduces the definition of the prototyping infrastructure for the use case, which includes both deployment diagrams for the whole use-case and a reference Java-based implementation for each choreography participant (i.e. Web service).

Keyword list

Use Case, scenario, requirement, Airport, architectural design, implementation, services, choreographies, demonstration, deployment diagram, BPMN2.0, SOAML, service architecture

Acronyms & Abbreviations

Item	Description
ATC	Air Traffic Control
BPMN	Business Process Modelling Notation
CDM	Collaborative Decision Making
DOW	Description Of Work
DPWS	Devices Profile for Web Services
FI	Future Internet
HTTP	Hypertext Transfer Protocol
MID	Mobile Internet Device
QoS	Quality of Service
TBD	To Be Defined
MID	Mobile Internet Device (e.g. laptops, smartphones)
OMG	Object Management Group
POI	Point Of Interest
REST	Representational state transfer
SOAML	Service oriented architecture Modeling Language
SOAP	Simple Object Access Protocol
SoTA	State of The Art
WS-*	All types (*) Web Service
WS-I	Web services Interoperability (e.g. “a WS-I profile”)
UML	Unified Modeling Language

Table of Contents

Table of Figures

1. Introduction

1.1. Purpose

The main purpose of this deliverable is to provide the description and **design of services** for the second SocEDA use-case termed “Air Traffic Flow Management”.

In order to lower the abstraction level of this deliverable, as well as move towards the upcoming use-case prototype to be delivered with D5.4.2 and D5.4.3, we conducted two complementary design approaches:

- **Choreographies design**: this was done by relying on the BPMN2.0 that gives a fine-grained representation of the service coordination that takes place in the use-case.
- **Services design**: we complement choreographies by adding SOAML (Service oriented architecture Modeling Language) models. SOAML is an open source specification project from the Object Management Group (OMG), describing a UML profile and metamodel for the modelling and design of services within a service-oriented architecture. These SOAML models bring lower-level information on how to design choreography participants (i.e. Web services). This information could not have been defined by relying only on BPMN2.0. For instance, SOAML models specify how services will interact at runtime, establishing specific message exchange patterns.

1.2. Challenges and advances over SOTA

The “Air Traffic Flow Management” use case is focused on improving the coordination between mostly **critical** airport actors and services (such as *airline*, *airline ground staff*, *air traffic control*, *Security Company*) in air airplane rerouting context. In this particular case coordination between these services already exists since it is crucial to a correct air traffic management. Still, there is room for improvement through systematic specification and automation in a domain where heterogeneous ad-hoc solutions are often used. We provide choreographies (see sections from 2.4.1 to 2.4.3) in this perspective.

The solution we provide using SocEDA deals with a challenge that is difficult to tackle with traditional service coordination approaches based on orchestrations: **coordination runtime responsibility**. Indeed, using an orchestration, one need to identify a central control point: the orchestrator. In this highly distributed airport context, where numerous actors are involved, none has the necessary prerogatives to incarnate this central point. Moreover, it would be very risky to have such a single point of failure. The solution to this issue is naturally provided by the choreography paradigm that is distributed by nature: there is no single responsibility point; the responsibility is dispatched to all the participants.

1.3. Scope

This document addresses, in the following sections, the precise definition of multiple use-case-related elements: *choreographies*, *service contracts* and *participants*.

The following chapter relies on the already introduced BPMN2.0 notation for definition of choreographies (sections 2.1), and SOAML for participants (sections 2.5).

Concerning contracts, SOAML defines a “Service Contract” as *“the specification of the agreement between providers and consumers of a service as to what information, products, assets, value and obligations will flow between the providers and consumers of that service – it specifies the service without regard for realization, capabilities or implementation.”* [1]. Yet a service contract does not specify its precise bounded providers and consumers at design time: it specifies ‘abstract’ roles that will subsequently be bound with actual participants (is this particular case, Web services) at runtime. This characteristic is required

by the **loose coupling** paradigm of SOA that is widely adopted by SocEDA. As such, for use-case implementation and demonstration purposes, this deliverable introduce definitions for these roles and the interfaces they provide or require in sections 2.5.

Finally, **chapter 3**, will give an early definition of the prototyping infrastructure of this use-case, based on UML deployment diagrams (section 3.1) to give an high-level overview of the deployment context of introduced services and choreographies.

2. "Air Traffic Flow Management" use-case

2.1. Overview description

The action takes place in a crisis context where a failure has been detected inside a nuclear plant. Scientists must be brought to place immediately. This use-case describes the various consequences that arise from rerouting airplanes due to an unexpected flight that has to take place from an airport. In that case, a reroute request is emitted for every flight currently reaching this airport and multiple well-known steps must be followed in order to find an alternative destination airport, keep passengers informed, as well as prepare for their unexpected arrival at the new airport. These are the steps that are described in the process shown in Figure 2. The textual details of these steps are given in Table 2.

It is to be noted that all the steps in the figure and table are **focused on one specific flight**, which means that each and every one of them need to be independently followed **for every flight** impacted by the rerouting requests.

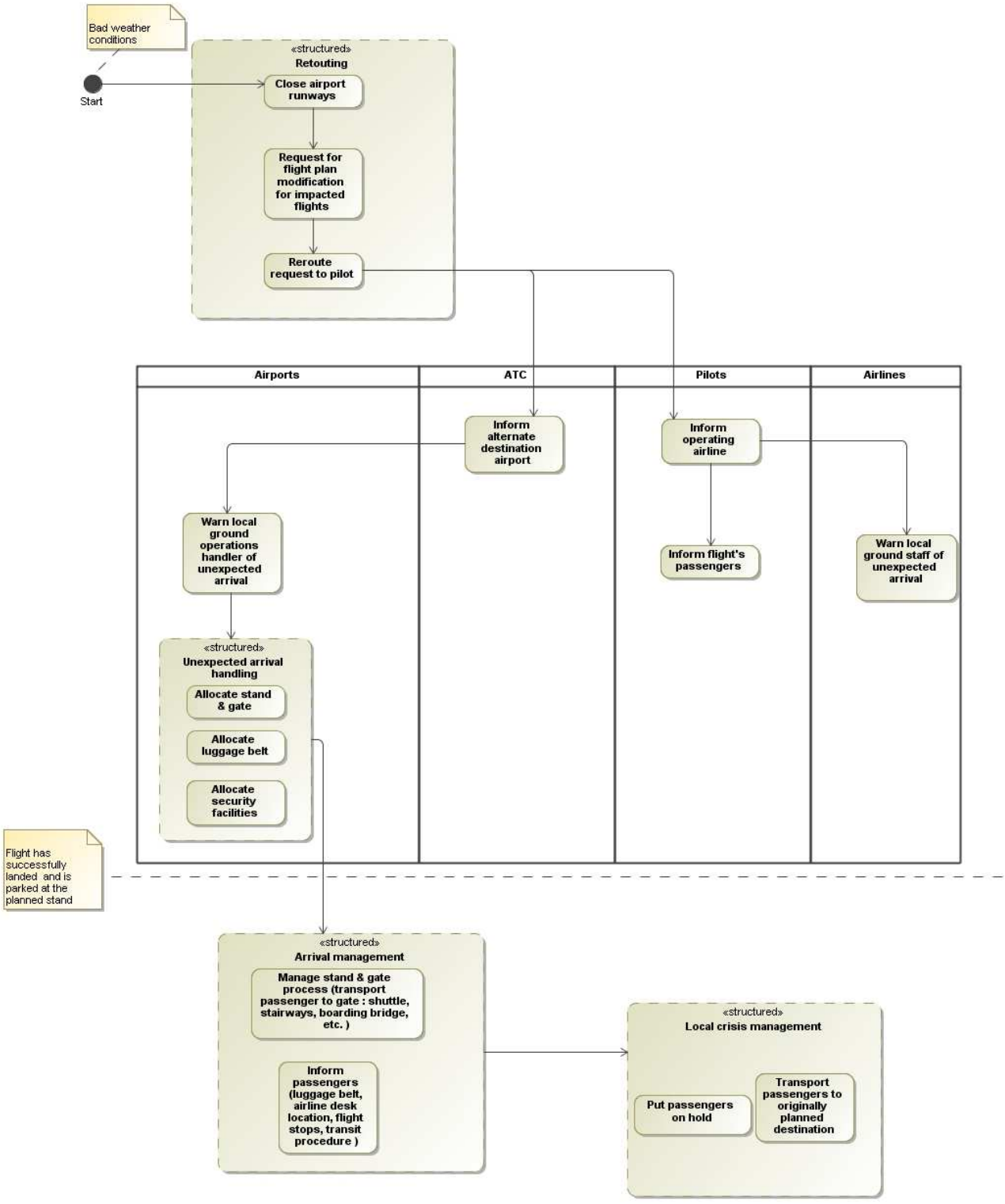


Figure 2 - Overview UML activity diagram of the use-case

Step	Description
Start	In a crisis context, an airport must make an airplane to take-off immediately and thus, prevent every incoming airplane from landing.
1: Rerouting	A decision is made to close runways and flight plans for all incoming scheduled flights need to be modified accordingly (step 2). Airport ATC communicates with pilots (of incoming flights scheduled for landing at

Step	Description
	this airport) for reroute request.
2: Flight plan modification	Once agreement is found on the final destination between ATC and the pilot: 1. The pilot informs his airline of the flight plan modification and gives new information to his passengers. 2. The alternate destination airport is informed of flight plan modification by ATC.
3: Unexpected arrival warning	The local destination Airport authority warns ground operations handlers (Airport Bus company, Luggage Handling company, Security company, etc.) of the unexpected arrival. The Airline informs warns its own local ground staff (at destination) of the arrival at their airport.
4: Unexpected arrival handling	The local destination Airport authority manages stand and gate, luggage belt and customs and security allocation, in anticipation of unexpected arrival.
5: Arrival management	The local destination Airport authority puts into place stand and gate management (to transport passengers by bus to gate if needed, etc.) and provides information to passengers such as luggage belt location.
6: Local crisis management	The Airline ground staff makes a decision between putting the passengers on hold (for the night) until the weather conditions are restored, or directly transporting the passengers to the originally planned destination (using new flight, bus and/or rail).

Table 2 - Individual steps of the "Bad weather at destination" scenario

2.2. Actors involved

2.2.1. People

- **Pilot**: the pilot is the one commanding a specific plane being rerouted.
- **Passengers**: from the perspective of a particular step of this scenario, passengers are the group of people contained in a specific plane that is being rerouted. They are treated as a single indivisible group of people and not independently.
- **Airline Ground Staff**: the alternative airport staff pertaining to the company of a specific plane being rerouted.
- **Security Agents**: deal with any security procedure the passengers are subjected to upon arrival at the alternative airport.
- **Luggage Handler**: alternative airport staff that has the responsibility of dealing with passengers baggage upon arrival at the alternative airport.

2.2.2. Services

Some of these services, and their descriptions, are shared with the previous scenario.

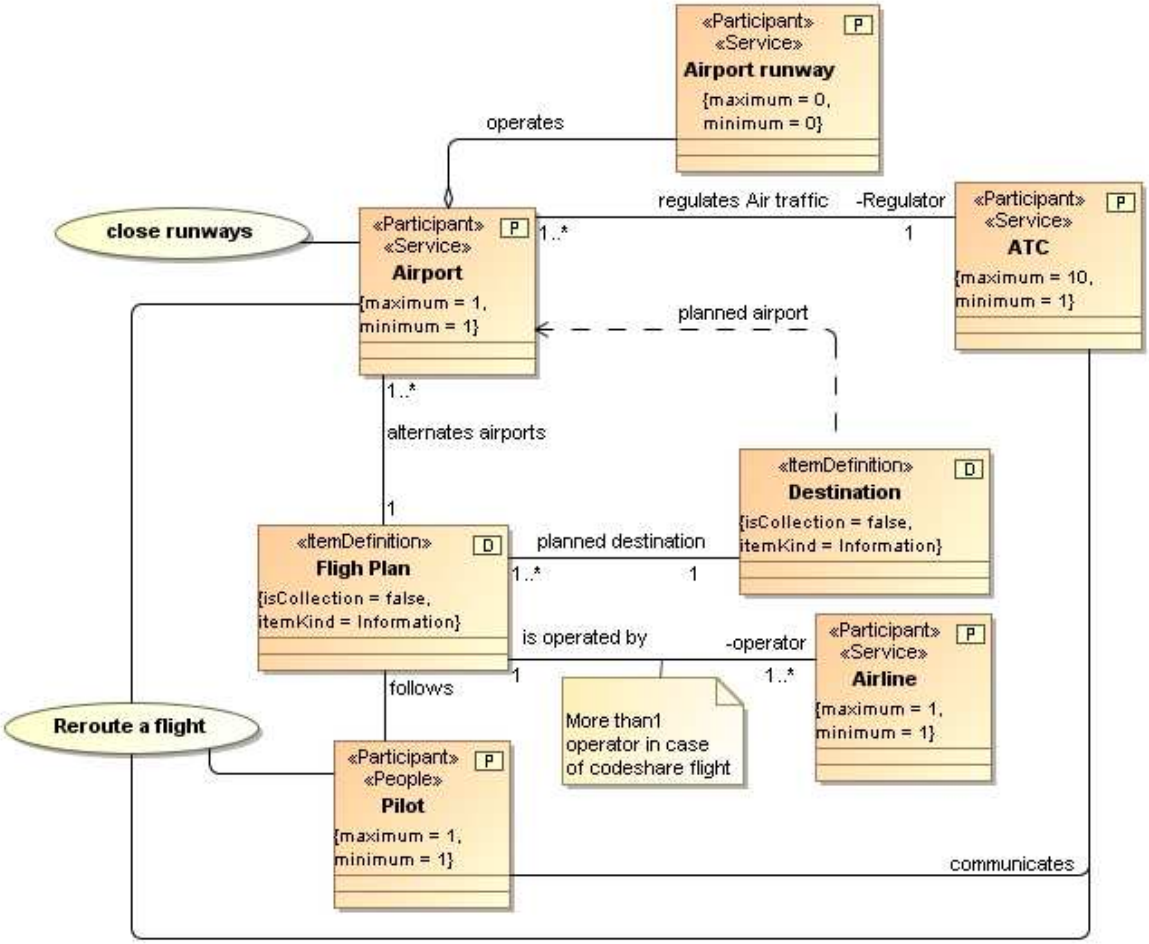
- **Airport**: service that represents the local airport authority and that is responsible for managing other partners in the terminals, such as the security company (e.g. it deals with allocating *security agents* to specific check-points depending on arrival and departures)
- **ATC (Airport Traffic Control)**: service that represents the authority that deals with airplane flight management and in this use-case that manages the rerouting of airplanes by informing pilots and airports.
- **Airline**: service used to retrieve all airline-related information and also inform airlines of airplanes rerouting.
- **Security Company**: service that represents the security company that manages all the *security agents*.
- **Airport Bus Company**: service that represents the company that manages all ground transportation by bus inside the airport perimeter (e.g. plane-to-gate and gate-to-plane operations).
- **Luggage Handling Company**: service that represents the company that manage all *luggage handlers* and assign them to the correct plane and/or luggage belt.
- **Weather forecast**: service used by *ATC* and *airport ground staff* to obtain weather forecasts.
- **Travel Agency**: service that represents the travel agency of a passenger. Used for communication with this travel agency.
- **Hotel**: service that represents the hotel where a passenger is assigned for the night. Used for communication with this hotel.
- **Ground Transportation Service**: service that represents the company that manages all ground transportation by bus outside the airport perimeter (e.g. to move passengers to their hotels).
- **Airport runway**: service that represents the runway managed by the airport and that need to be closed due to bad weather.

2.3. Step-level requirements

Requirements elicited from this chapter relates from the one defined in the document D1.1.1 – Requirements. Requirements prefixed with “UC2” are requirements to be covered by the use-case and also defined in the D1.1.1.

For requirements type explanation (eg SR, IR, etc.), please refer to the deliverable D1.1.1.

1: Rerouting

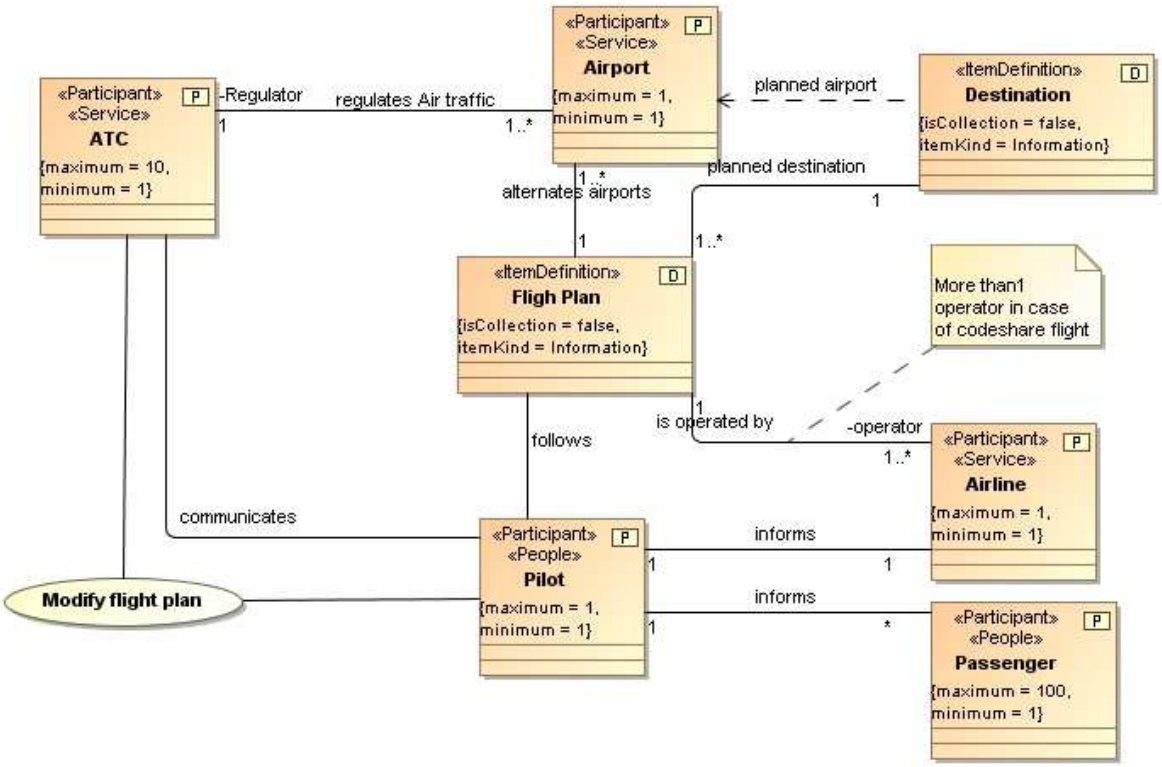


+ SR001 - Confidentiality
+ SR002 - Dynamic access control to services
+ SR003 - Circle of trust
+ SR004 - Integrity
+ IR003 - Human Interaction
+ ScR001 - Web scale support
+ ScR002 - Large Scale workflow support
+ ScR004 - Service request scalability
+ ScR005 - Events exchange scalability

UC2-PR002 - Collaborative decision making (CDM)
UC2-PR003 - ATFM CDM information propagation times
UC2-PR004 - Reroute request delay
UC2-PR005 - Delay to inform all impacted planes of ATFM
UC2-PR006 - Delay to close runways

UC2-AvR001 - Flight plan access by ground-based controllers
UC2-ScR003 - Plane rerouting

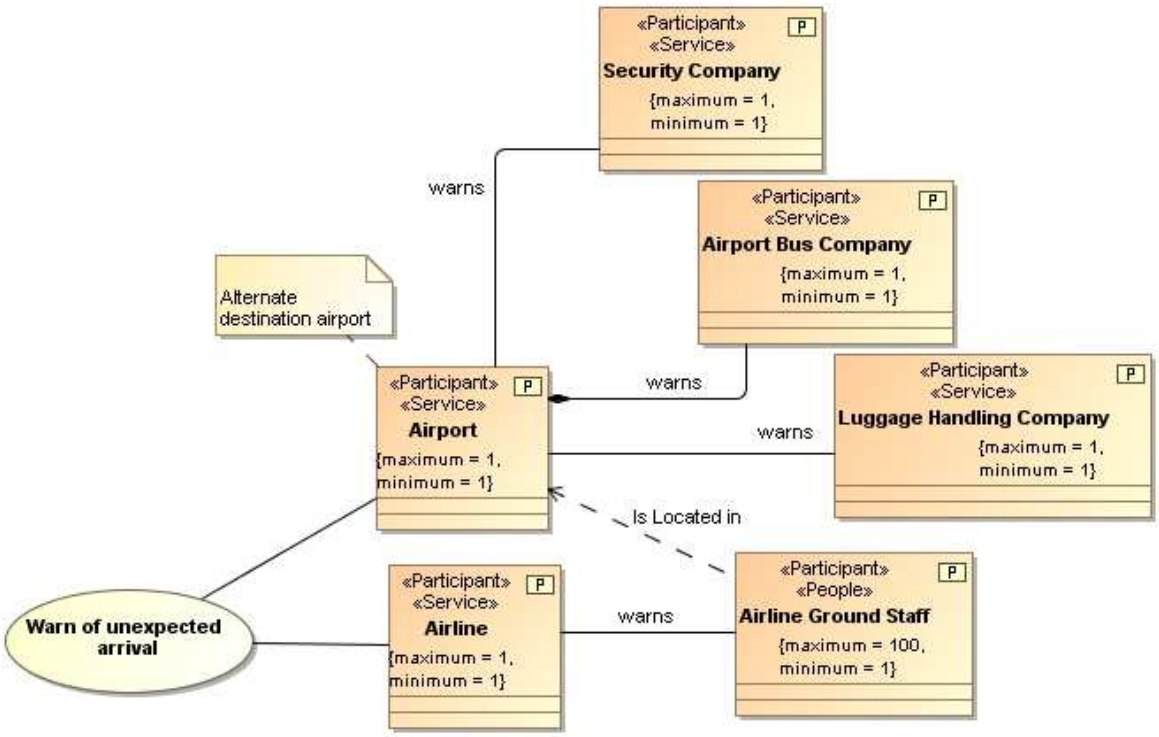
2: Flight plan modification



+ SR001 - Confidentiality
+ SR002 - Dynamic access control to services
+ SR003 - Circle of trust
+ SR004 - Integrity
+ IR003 - Human Interaction
+ ScR001 - Web scale support
+ ScR002 - Large Scale workflow support
+ ScR004 - Service request scalability
+ ScR005 - Events exchange scalability

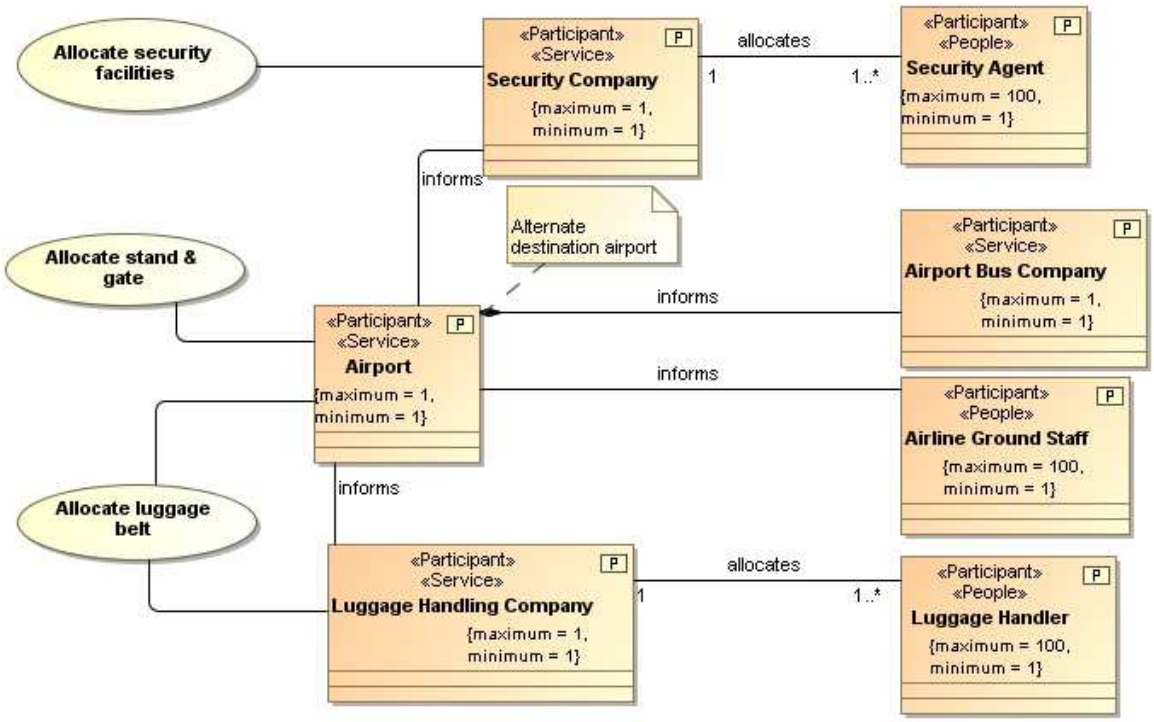
UC2-PR007 - Reroute agreement delay
UC2-PR008 - Flight plan modification agreement broadcasting delay
UC2-AvR002 - Alternate airport list availability
UC2-AcR001 - Rerouted plane arrival time accuracy
UC2-IR001 - Communication channels interoperability
UC2-RR001 - Communication channels reliability

3: Unexpected arrival warning



+ SR001 - Confidentiality
+ SR003 - Circle of trust
+ SR004 - Integrity
+ IR002 - Dynamic service discovery
+ IR004 - Runtime flexibility
+ ScR001 - Web scale support
+ ScR002 - Large Scale workflow support
+ ScR004 - Service request scalability
+ ScR005 - Events exchange scalability

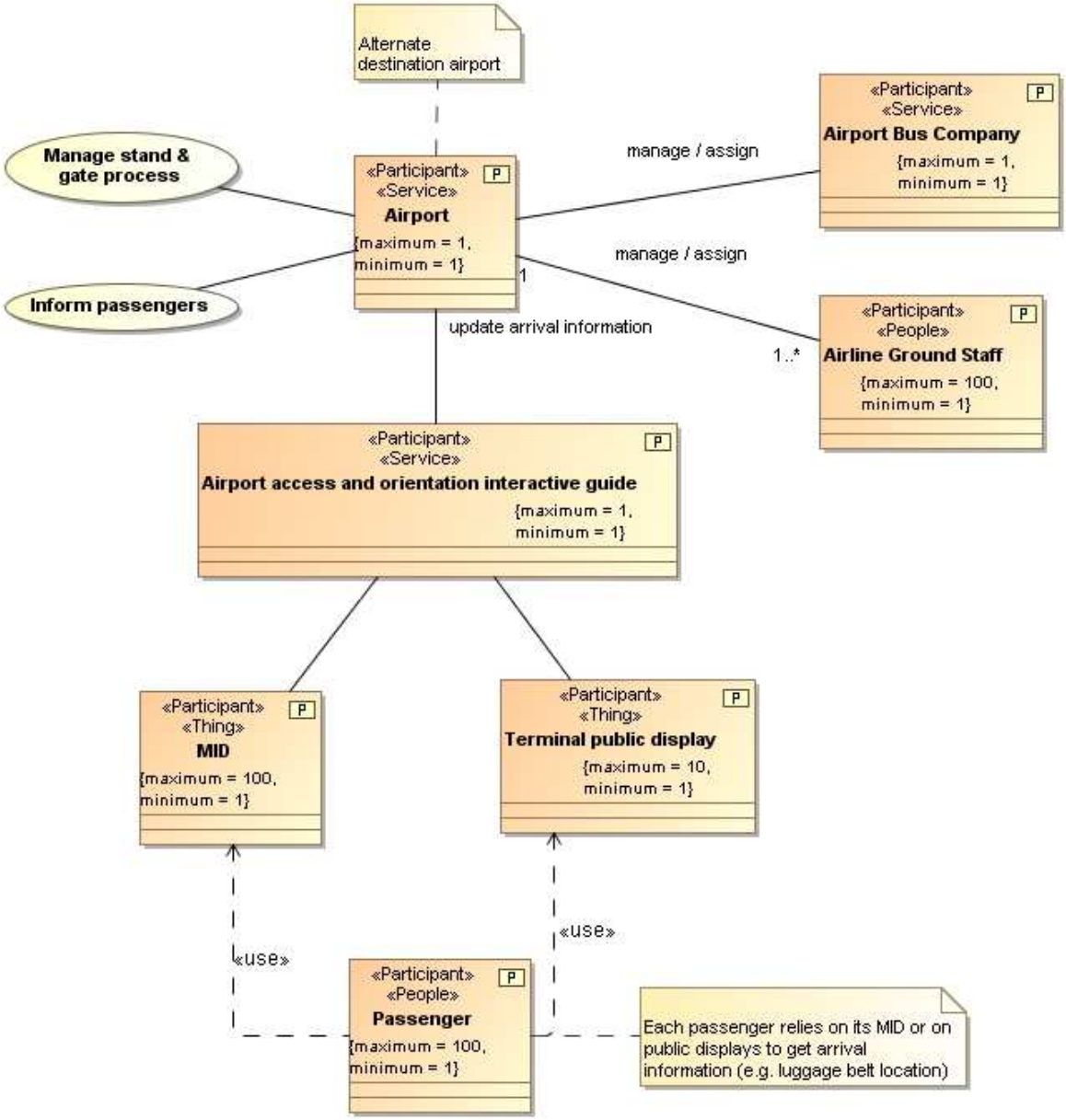
4: Unexpected arrival handling



+ SR001 - Confidentiality
+ SR003 - Circle of trust
+ SR004 - Integrity
+ IR002 - Dynamic service discovery
+ IR004 - Runtime flexibility
+ ScR001 - Web scale support
+ ScR002 - Large Scale workflow support
+ ScR004 - Service request scalability
+ ScR005 - Events exchange scalability

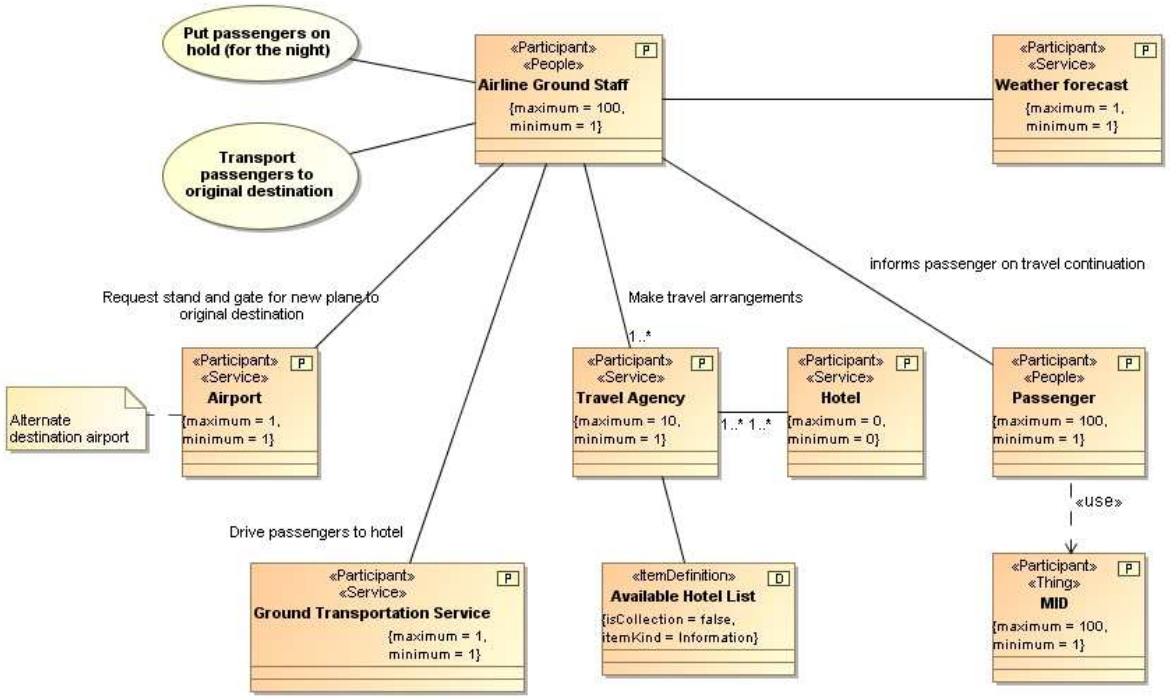
UC2-AvR003- Aiports congestion status availability
UC2-AdR003 - Adaptation to airports congestion (related to UC2-FR001)

5: Arrival handling



+ SR001 - Confidentiality
+ SR003 - Dynamic access to passenger data
+ SR004 - Circle of trust
+ SR005 - Integrity
+ IR001 - Highly heterogeneous services support
+ IR002 - Dynamic service discovery
+ IR004 - Runtime flexibility
+ ScR001 - Web scale support
+ ScR002 - Large Scale workflow support
+ ScR004 - Service request scalability
+ ScR005 - Events exchange scalability
+ PR001 - Run of multiple parallel workflows among several web-services

6: Local crisis management



+ SR001 - Confidentiality
+ SR003 - Dynamic access to passenger data
+ SR004 - Circle of trust
+ SR005 - Integrity
+ IR001 - Highly heterogeneous services support
+ IR002 - Dynamic service discovery
+ IR004 - Runtime flexibility
+ ScR001 - Web scale support
+ ScR002 - Large Scale workflow support
+ ScR004 - Service request scalability
+ ScR005 - Events exchange scalability
+ FR007 - Platform features as Web services
+ PR001 - Run of multiple parallel workflows among several web-services

2.4. BPMN2.0 Choreographies

We introduce hereafter the choreographies dedicated for managing the unexpected arrival of passengers at an alternative airport.

2.4.1. Choreography a - Rerouting

This choreography includes all the tasks pertaining to the rerouting activity but dispatches to the two following choreographies the ones related to managing the unexpected arrival (see section 2.4.2) and arrival handling (see section 2.4.3).

This choreography puts multiple actors in the spotlight, including *pilot*, *airline ground staff*, *security agents*, *luggage handlers*, etc.

Another important aspect of the following choreography, including its sub-choreography, is that, for simplification purpose, we focus on the choreography as viewed from the specific angle of one particular flight. As such, there will be **as many instances of this global choreography in the system as there are flights to be rerouted**.

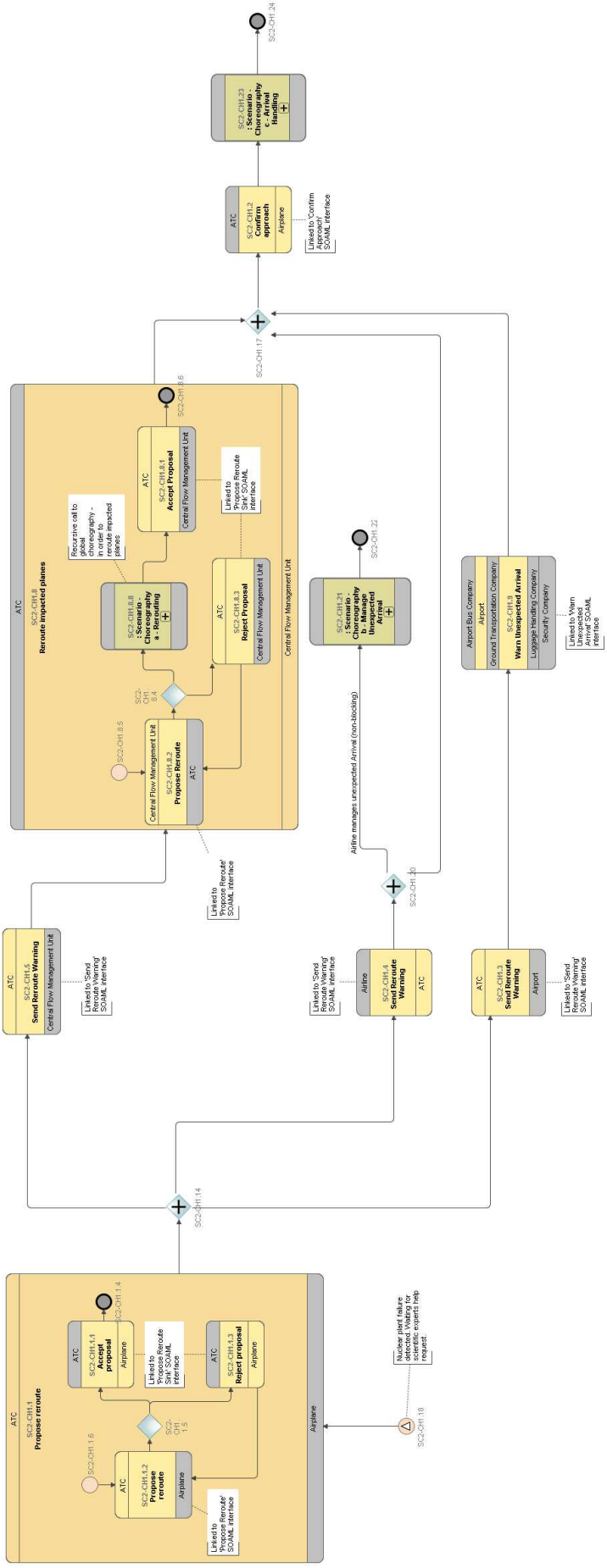


Figure 1: Choreography a - Rerouting

2.4.2. Choreography b - Manage Unexpected Arrival

This particular sub-choreography gives the details pertaining to the management of unexpected arrival of passengers at an alternative airport. It includes the pre-emptive decision made by airline ground staff of travel continuation modalities for rerouted passengers, based on the information “special flight has taken-off, airport is available again” gathered from the original destination airport.

In fact, the airline ground staff needs to decide if it should directly prepare a new flight to their original destination or put them on hold for the night. Both cases imply logistic consequences: for instance, putting passenger on hold for the night means that information will need to be exchanged with travel agencies of passengers, hotels will be booked for one night and, finally, ground transportation will need to be allotted in order to convey passengers to these hotels – the main idea being to put everything in place before plane landing and passenger arrival at the airport

In this choregraphy, the particularity is that there are partners (called *amenities*) that are not directly shown in the choreography diagram but are dynamically discovered at runtime, in order to prepare a new flight to direct passengers to their original destination. These services are *Airplane*, *Airport Bus Company* (for transportation inside the airport compound), *Luggage Handling Company* and *Security Company*. **They are dynamically retrieved at runtime**, through Airline, Stand and gate management and Airport services in tasks SC2-CH2.6.2 to SC2-CH2.6.7. As such, in this use-case centred choreography, **late binding** is not only provided transparently by the SocEDA runtime for any task, it is also **explicitly stated in the choreography itself**. Please note that this is only possible because we rely on Web services that are able to return other Web services endpoints (here of amenities) when requested.

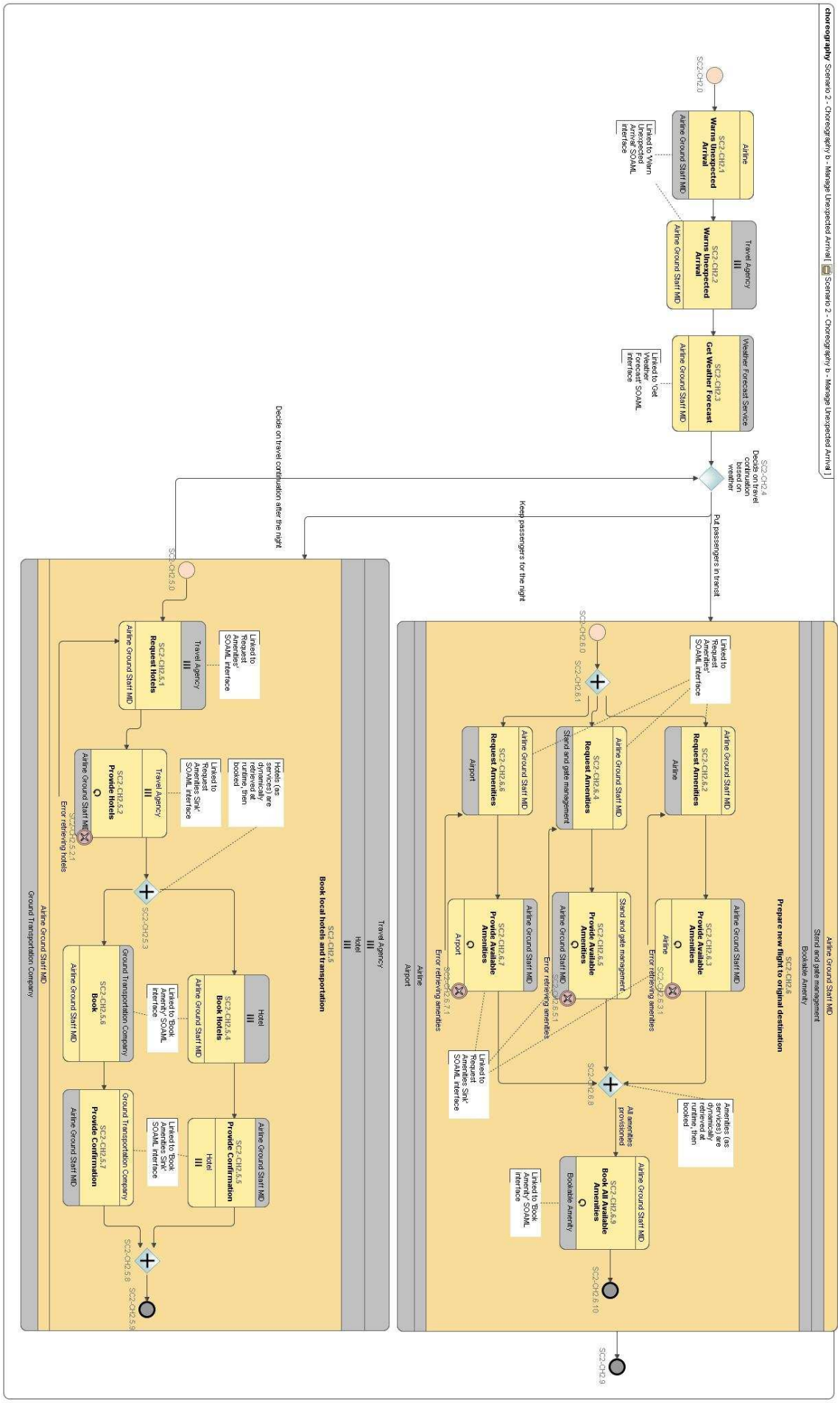


Figure 2: Choreography b - Manage Unexpected Arrival

2.4.3. Choreography c - Arrival Handling

This choreography deals with handling the arrival of passengers at alternate destination: this includes reservation of amenities, as in previous choreographies, for managing the arrival of the plane: *stand and gate management* amenities that are dynamically retrieved at runtime, as well as *luggage handling company*, *Security Company* and *airport Bus Company*.

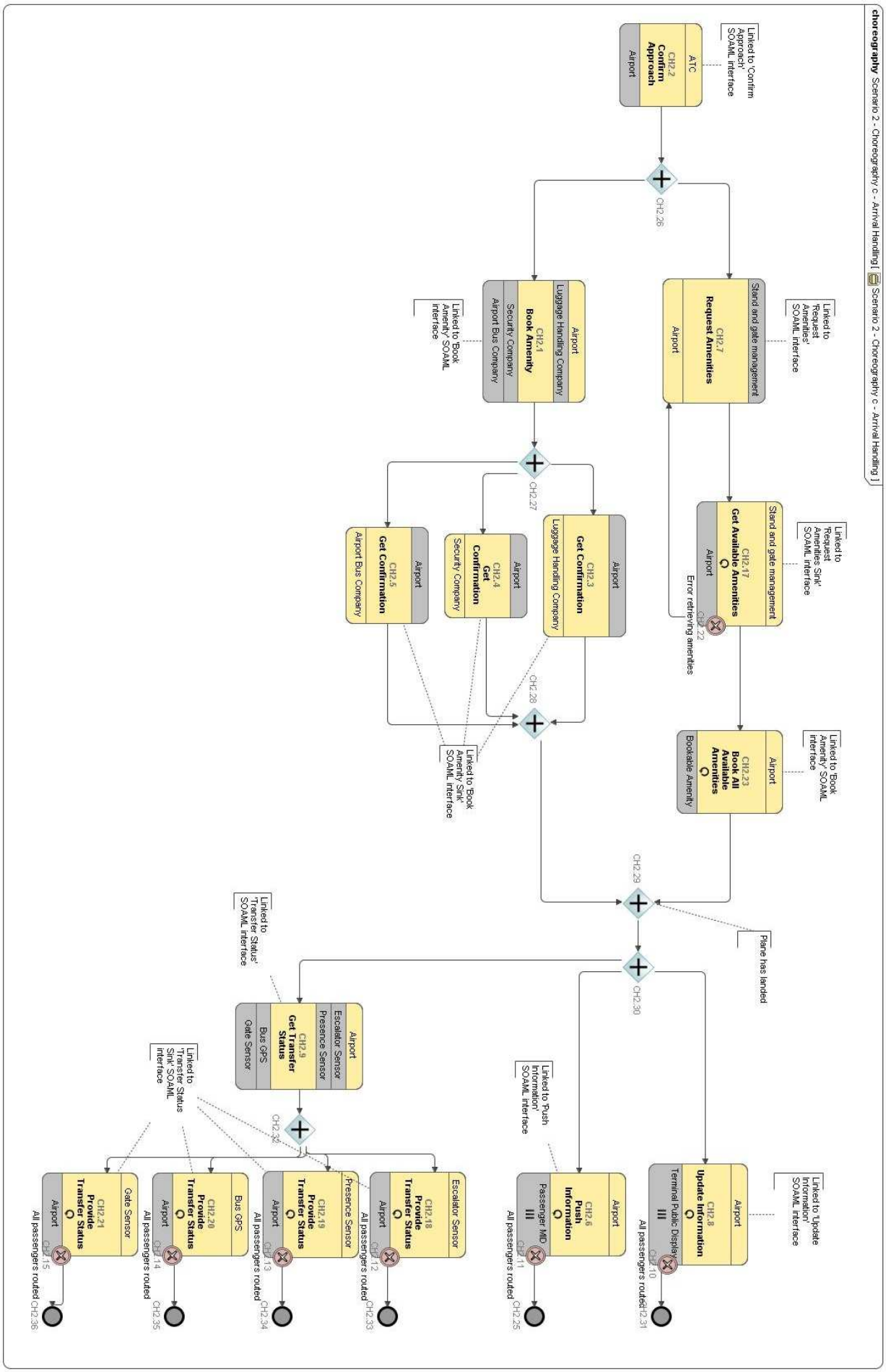


Figure 3: Choreography c - Arrival Handling

2.5. SOAML Participants

In this section we outline service **Participants**. According to SOAML specification: “Participants are either specific entities or kinds of entities that provide or use services. Participants can represent people, organizations or information system components. Participants may provide any number of services and may consume any number of services. Participants provide or consume services via **Ports**. A port is the part or feature of a participant that is the interaction point for a service – where it is provided or consumed. A port where a service is offered may be designated as **a «Service» port** and the port where a service is consumed may be designated as **a «Request» port**”.

To illustrate Participants present in this scenario we use Composite Application Components Diagrams. For each Participant we denote the types of «Service» and/or «Request» Ports using SOAML notations. Types are specific Service Interfaces that each Participant must implement to adhere to Service Contracts that he participates either as a provider or as a consumer. Participants follow a “matching service interface” pattern, meaning that for every provided Request Port on a Participant there must exist at least one compatible (of the same type) Service Port on some other Participant.

2.5.1. Airline

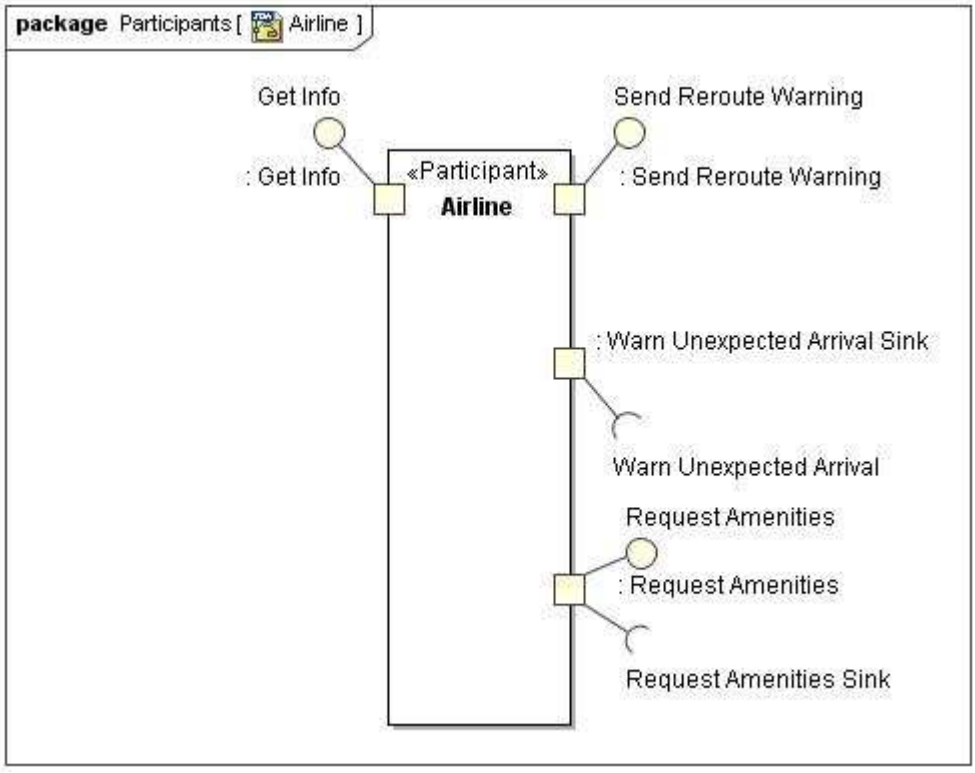


Figure 4: Participant Diagram Airline

2.5.2. Airline Contractor MID

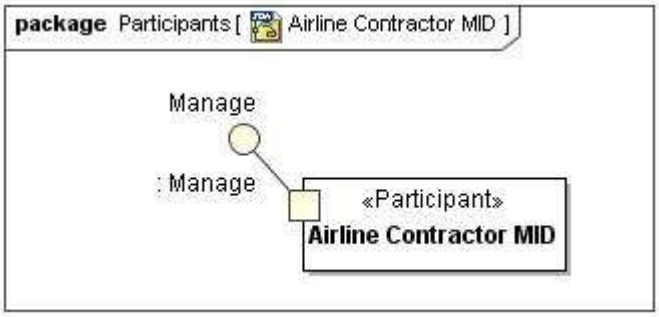


Figure 5: Participant Diagram Airline Contractor MID

2.5.3. Airplane

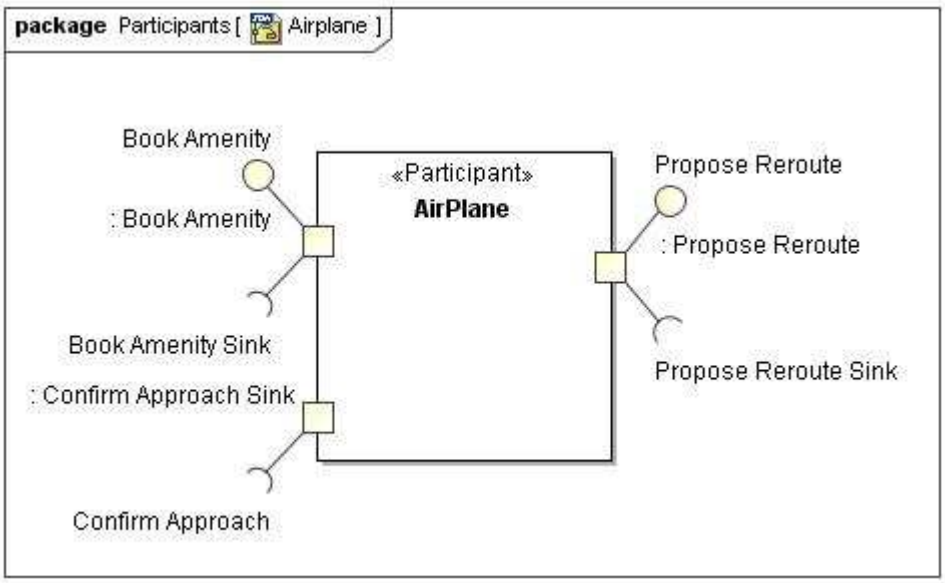


Figure 6: Participant Diagram Airplane

2.5.4. Airport

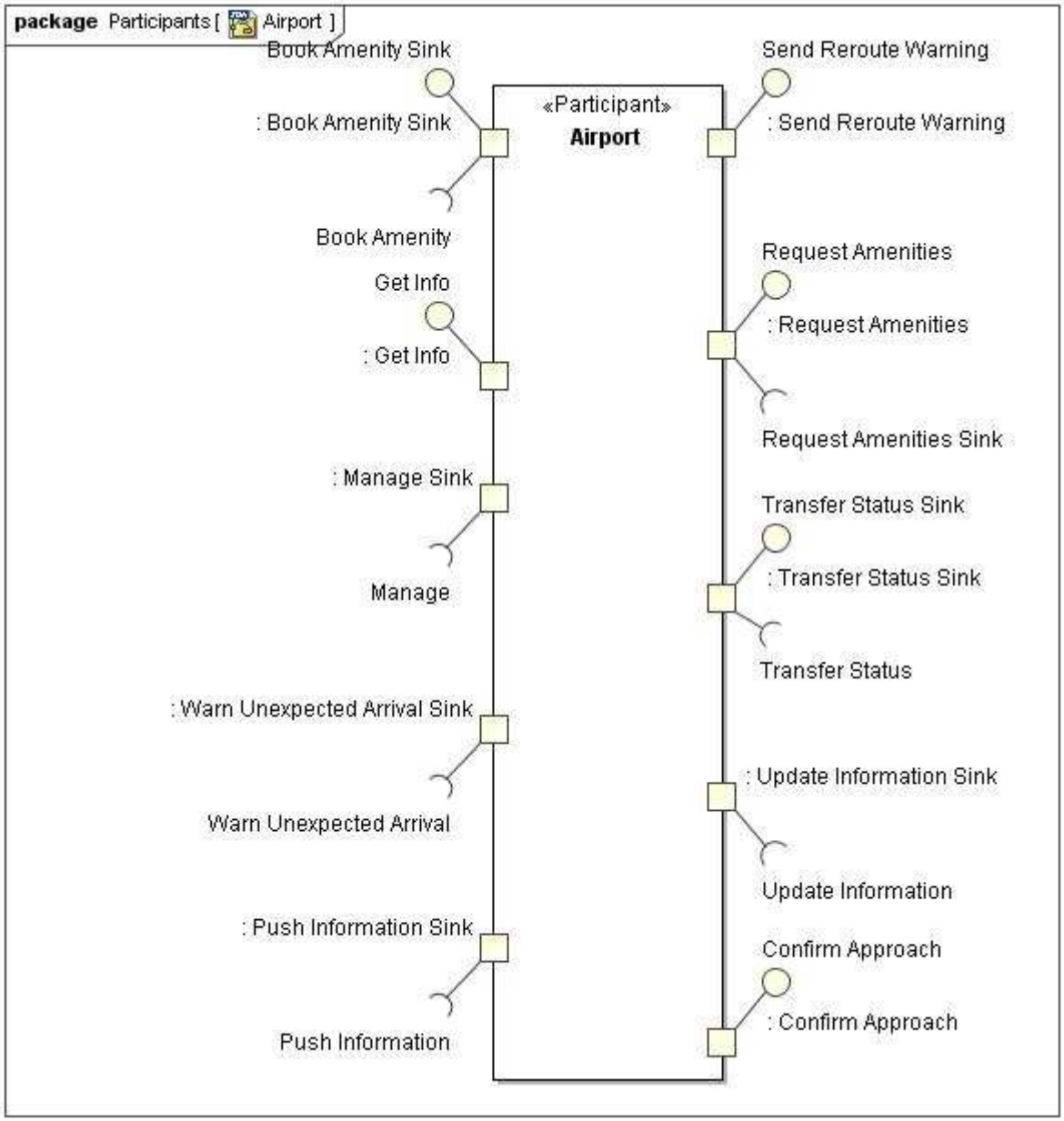


Figure 7: Participant Diagram Airport

2.5.5. Air Traffic Control

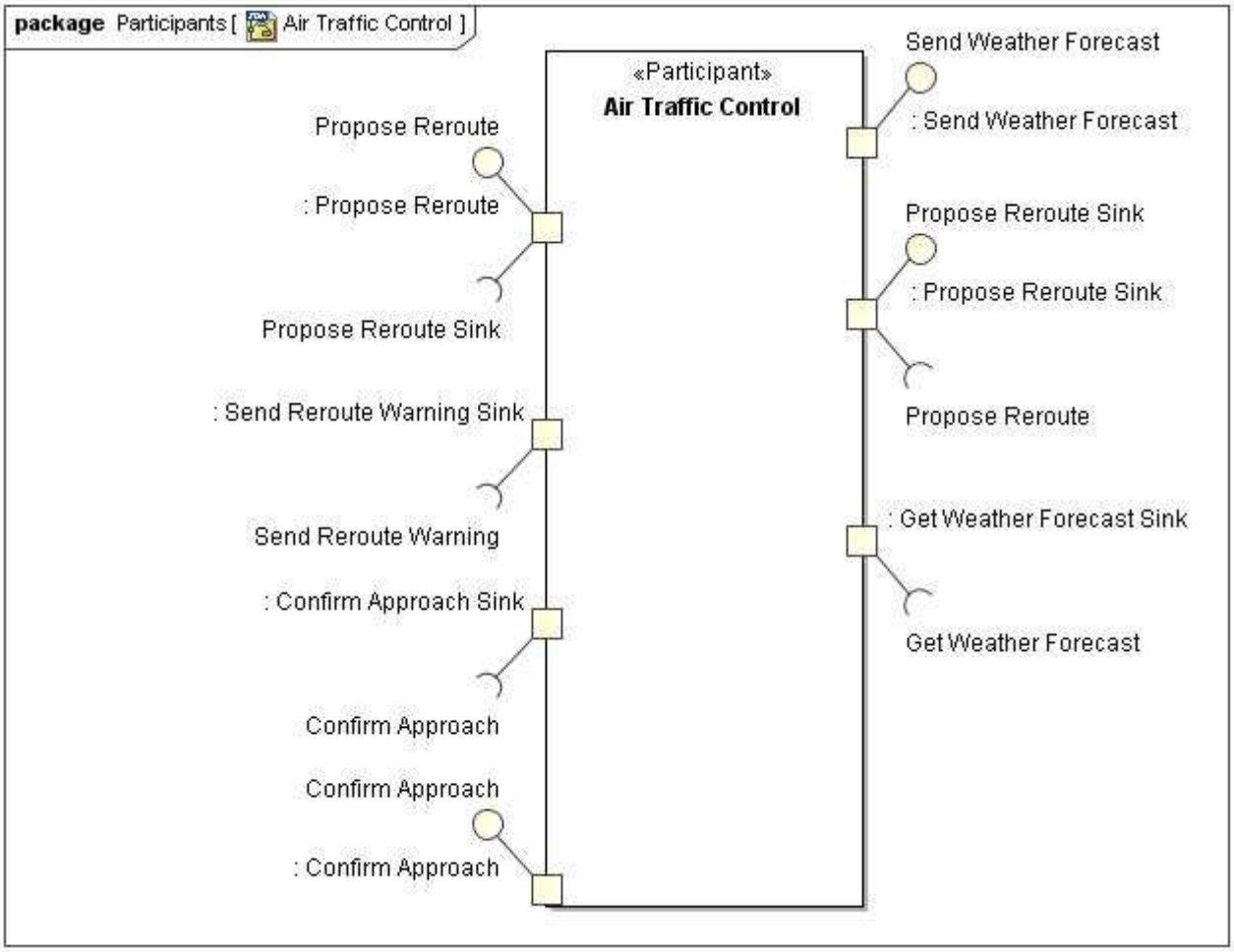


Figure 8: Participant Diagram Air Traffic Control

2.5.6. Bookable Amenity

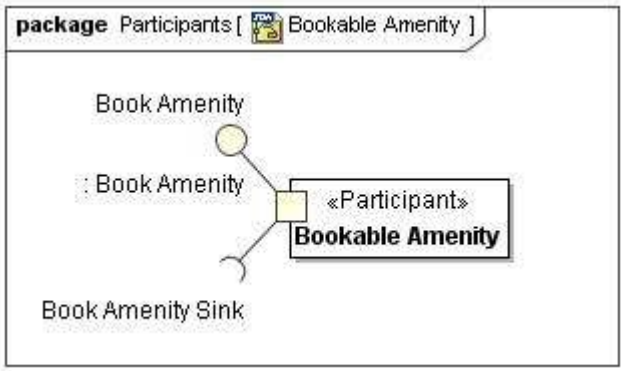


Figure 9: Participant Diagram Bookable Amenity

2.5.7. Central Flow Management Unit

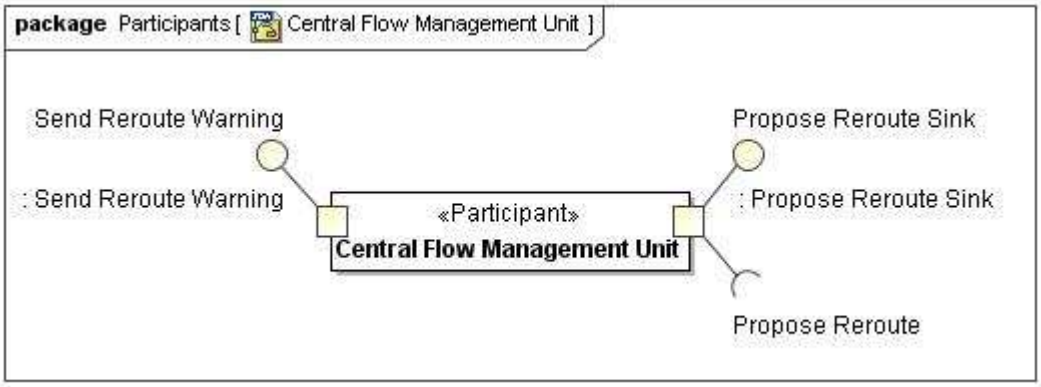


Figure 10: Participant Diagram Central Flow Management Unit

2.5.8. Stand and gate management

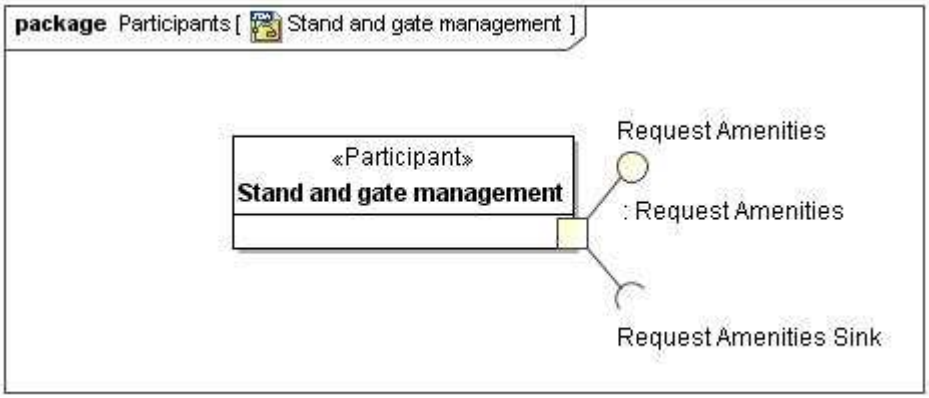


Figure 11: Participant Diagram Stand and gate management

2.5.9. Amenity

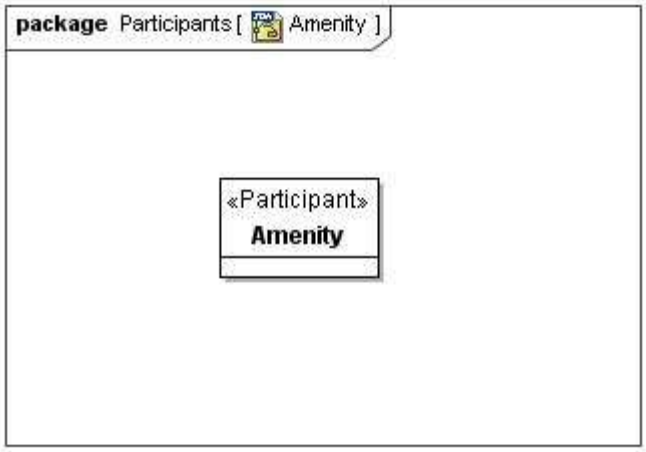


Figure 12: Participant Diagram Amenity

2.5.10. Airport Bus Company

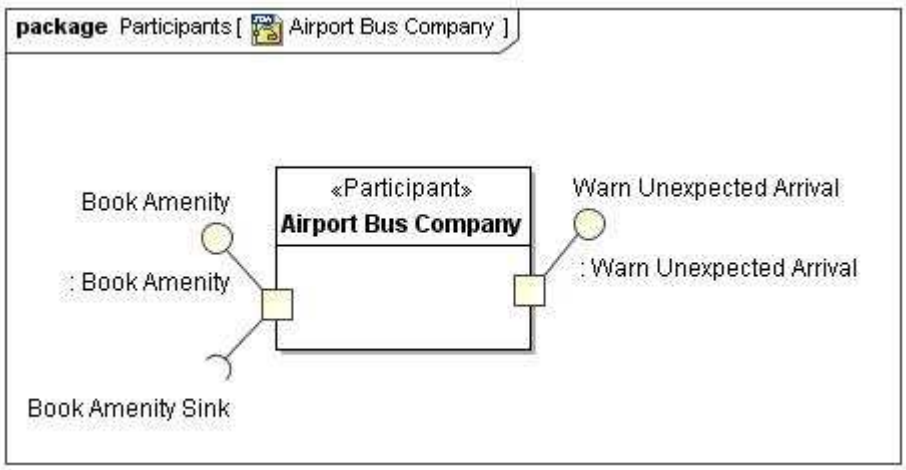


Figure 13: Participant Diagram Airport Bus Company

2.5.11. Ground Transportation Company

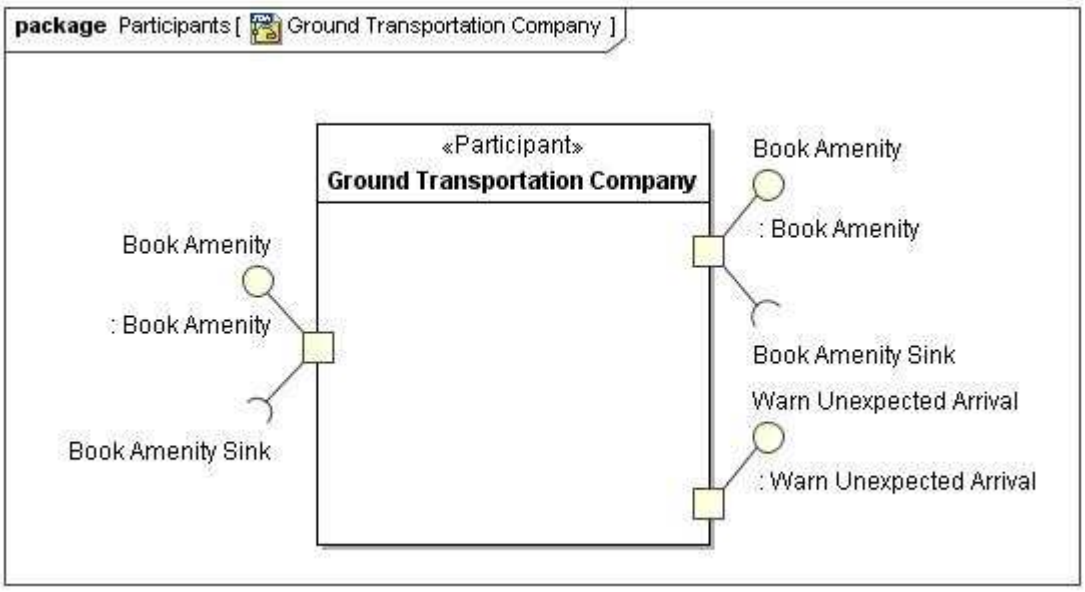


Figure 14: Participant Diagram Ground Transportation Company

2.5.12. Luggage Handling Company

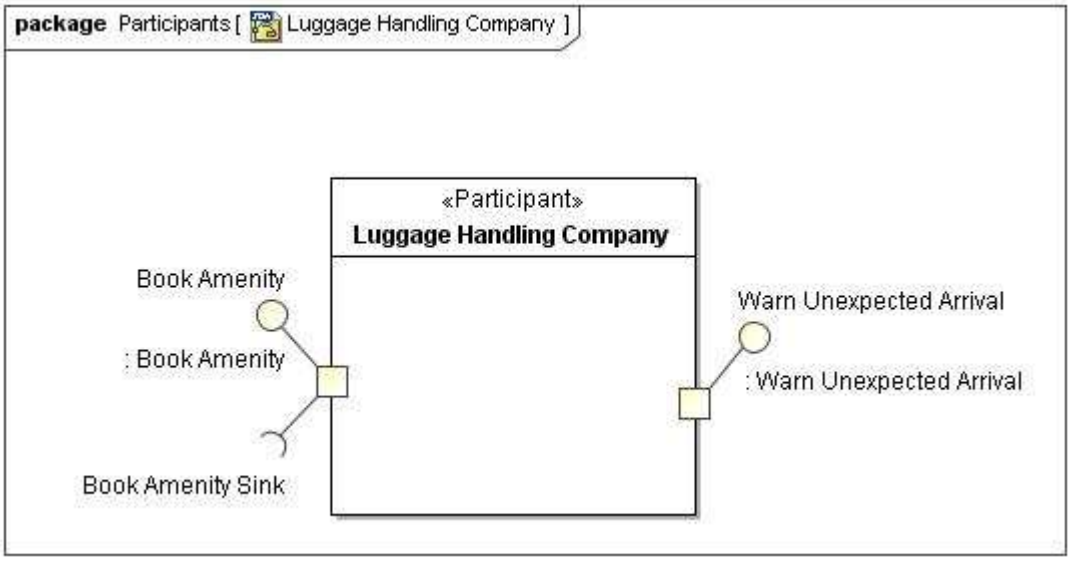


Figure 15: Participant Diagram Luggage Handling Company

2.5.13. Security Company

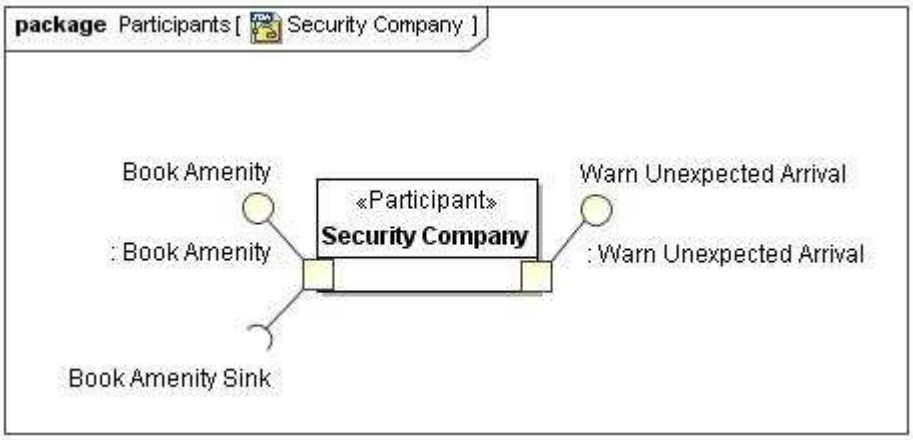


Figure 16: Participant Diagram Security Company

2.5.14. DemoParticipant

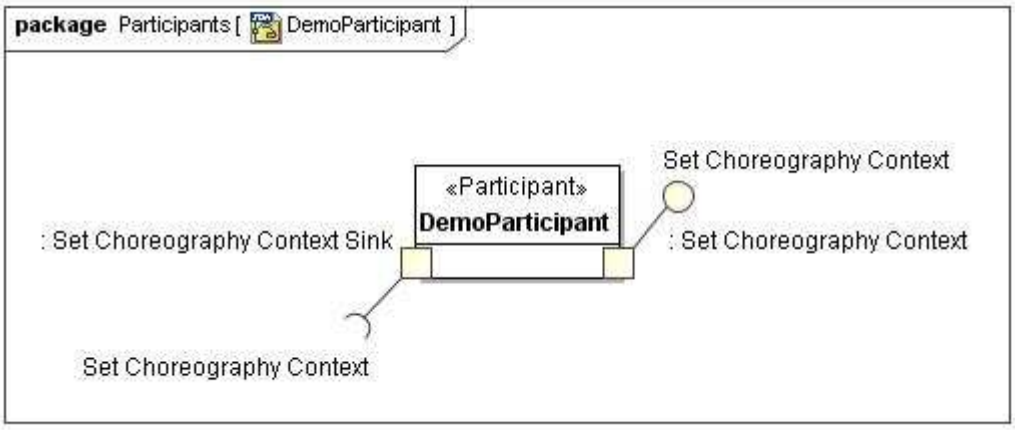


Figure 17: Participant Diagram DemoParticipant

2.5.15. Participant Hierarchies

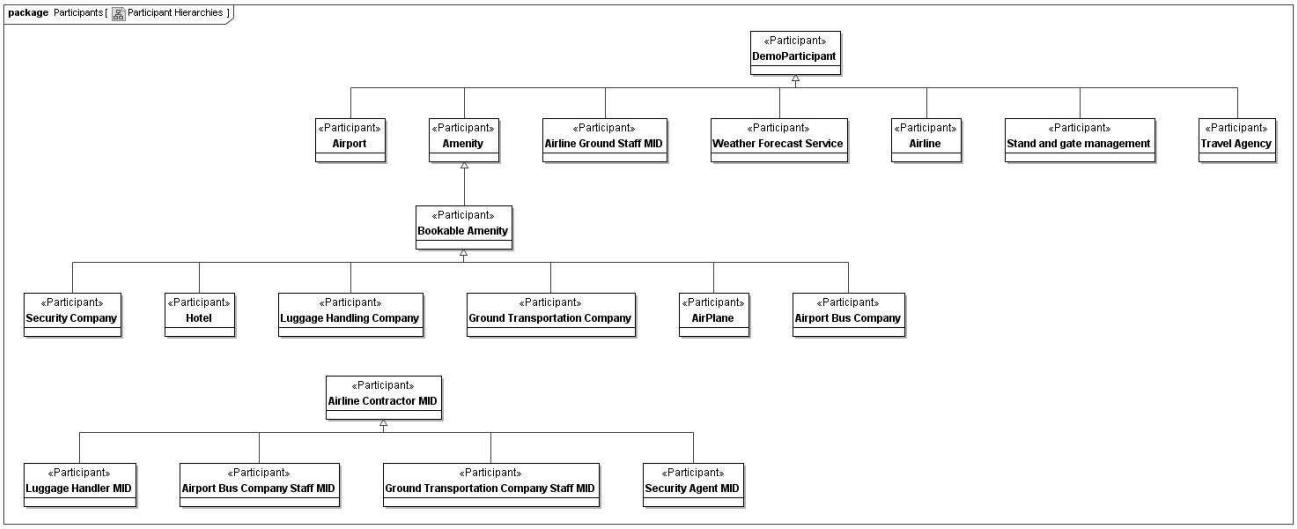


Figure 18: Participant Hierarchies

3. Definition of the prototyping infrastructure

3.1. UML Deployment diagrams

This section describes the static deployment view of the systems. UML Deployment diagrams are here used to visualize the topology of the physical components of a system where the software components are deployed. It consists of nodes and the communication path among them.

The network topology is organized around five networks:

- The Internet, from which can be accessed an offshore cloud hosting the utility companies, the airport cloud and the airline proxy in the Air Traffic Control (ATC) infrastructure;
- The Airport intranet, from which can be accessed the airport cloud, all the devices residing in the airport as well as the Air Traffic Control (ATC);
- The CFMU network connecting the ATCs with the Central Flow Management Unit (CFMU);
- The Controller-Pilot Data Link Communications (CPDL) connecting the ATCs and Airlines with the airplanes.

Again, nodes and devices are plugged on these networks:

- The Offshore Cloud still gathers and hosts the “utility companies”, i.e. the Ground Transportation Companies, the Hotels, the Travel agencies, the Weather Forecast Services and the Airline Companies, which take advantage of the Cloud infrastructure to cope with their scalability issues;
- The Airport Cloud hosts the main services of the airport – Airport Service to Passenger, Airport Map Services, Airport Flight Services and the Airport C2 System – as well as the security company services, the Airport Bus Companies and Luggage Handling Companies – the last three need to be within the airport infrastructure –;
- The Air Traffic Control Cloud hosts airline proxies to the ATC, ATCs and the CFMU – the cloud infrastructure allows these agency to share their IT resources –;
- The airplanes.

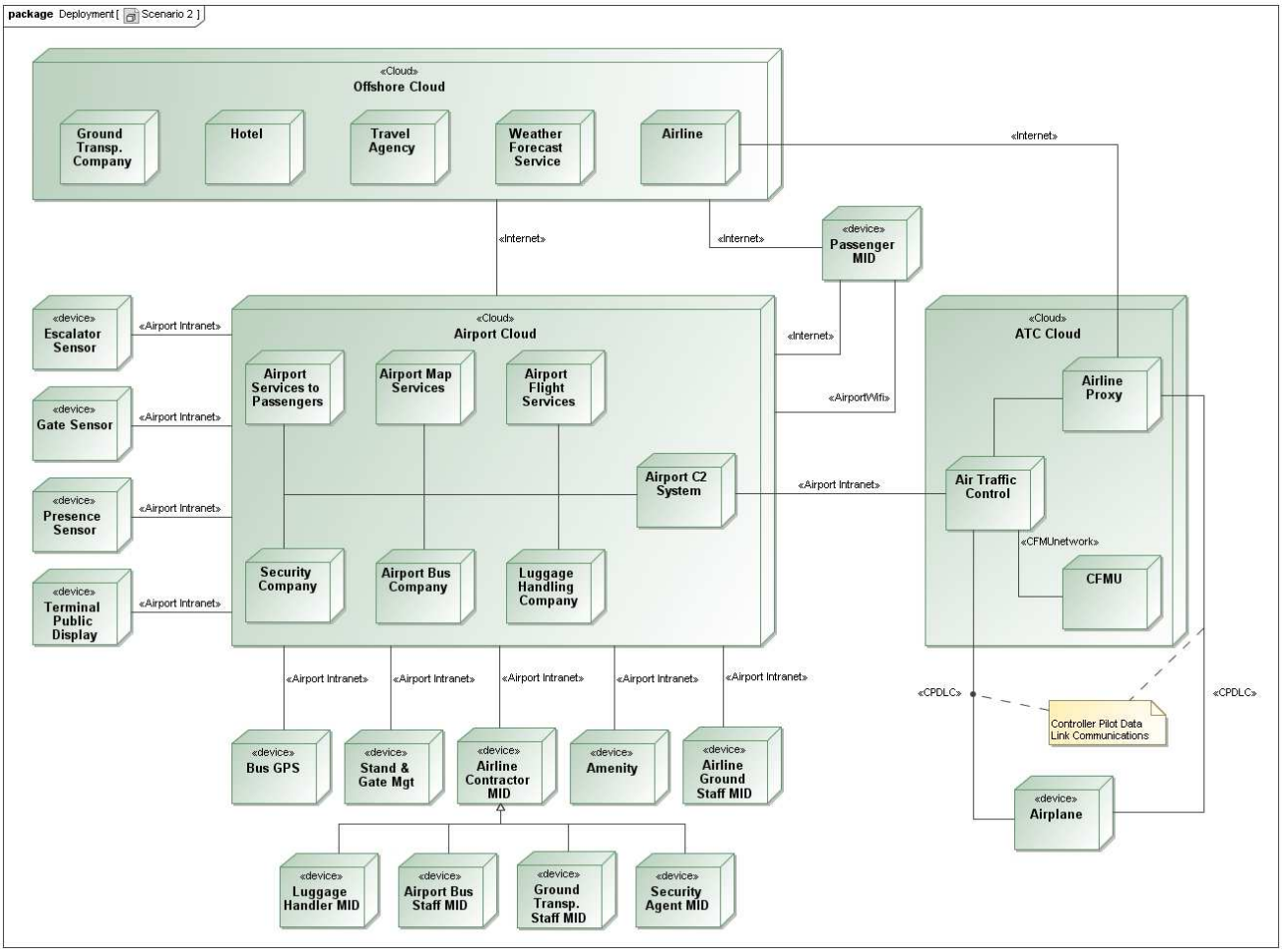


Figure 19 - Deployment diagram for the scenario

Each of these nodes host the matching component depicted in SOAML Participant section 3.3 except for the Airport participant which was spread on multiple nodes for scalability reasons; the following table shows the mapping airport service to SOAML port (within the Airport participant).

Airport Service	SOAML port (cf Section 3.3)
Airport Services to Passengers	Book Amenity
Airport Flight Services	Get Info
Airport C2 Systems	Manage
Airport C2 Systems	Warn Unexpected Arrival
Airport Services to Passengers	Push Information
Airport C2 Systems	Send Reroute Warning
Airport Services to Passengers	Request Amenities
Airport C2 Systems	Transfer Status
Airport C2 Systems	Update Information
Airport C2 Systems	Confirm Approach

4. Glossary

Elicitation	“The discovery, gathering, or ‘capture’ of requirements, often by developing scenarios.”

Scenario	An instance of a use case, expressed as a sequence of event . Holbatz describes a scenario as a “sequence of steps that defines a task performed to achieve an intent” [2]
Stakeholder	Someone with an interest in the future system who might have requirements on the system.
Step	“An action or event representing an atomic component of a scenario. In a use case, each step is normally described in a separate paragraph of text (...)” [2]