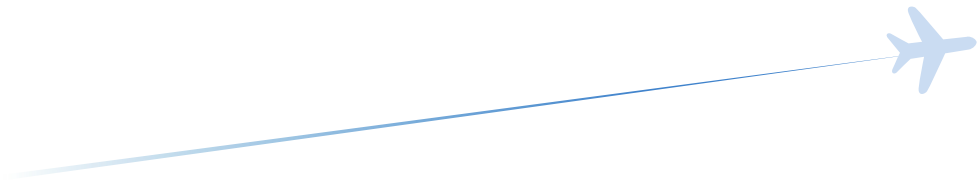

D2.1 Requirements Specification

Deliverable ID:	**D2.1**
Dissemination Level:	**PU**
Project Acronym:	**SlotMachine**
Grant:	**890456**
Call:	**H2020-SESAR-2019-2**
Topic:	**SESAR-ER4-27-2019 Future ATM Architecture**
Consortium Coordinator:	**Frequentis**
Edition Date:	**22 October 2021**
Edition:	**01.02.02**
Template Edition:	**02.00.02**

Founding Members

EUROPEAN UNION EUROCONTROL

SESAR JOINT UNDERTAKING

Authoring & Approval

Authors of the document

Name/Beneficiary	Position/Title	Date
Christoph Schuetz / JKU	WP 2 & 4 Leader	2021-06-24
Thomas Lorünser / AIT	WP 3 Leader	2021-06-24
Thomas Obritzhauser / FRQ	Senior Expert	2021-06-18
Christoph Fabianek / FRQ	Technical Manager	2021-06-12
Eduard Gringinger / FRQ	Project Manager	2021-10-22

Reviewers internal to the project

Name/Beneficiary	Position/Title	Date
Marie Carré / SWISS	WP 5 Leader	2021-07-01
Nadine Pilon / ECTL	WP 2 Co-Leader	2021-07-06
Thomas Obritzhauser / FRQ	Senior Expert	2021-07-05
Eduard Gringinger / FRQ	Project Manager	2021-07-02

Approved for submission to the SJU By - Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date
Christoph Fabianek / Frequentis	Technical Manager	2021-07-12
Eduard Gringinger / Frequentis	Project Manager	2021-07-12
Marie Carré / SWISS	WP 5 Leader	2021-07-12
Nadine Pilon / EUROCONTROL	WP2 Co-Leader	2021-07-16 (silent approval)
Christoph Schuetz / JKU	WP2, 3 Leader	2021-07-16
Lorünser Thomas / AIT	WP4 Leader	2021-07-16 (silent approval)

Rejected By - Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date
-	-	-

Document History

Edition	Date	Status	Author	Justification
01.00.00	2020-11-05	First draft	C. Fabianek	Initial draft
01.01.00	2021-02-01	Structure complete	C. Fabianek	document structure agreed
01.01.01	2021-03-26	Requirements complete	C. Schütz, T. Lorünser	internal review
01.01.02	2021-06-24	Internal release	C. Fabianek, C. Schütz, T. Lorünser	Ready for approval
01.01.03	2021-07-28	Final version	C. Fabianek, E. Gringinger	Release ready to deliver to SJU
01.02.01	2021-10-08	Revised version after review	T. Obritzhauser, E. Gringinger, C. Schütz, T. Lorünser	Ready for resubmission
01.02.02	2021-10-22	Revised version after review round 2	T. Obritzhauser, E. Gringinger, C. Schütz, T. Lorünser	Ready for resubmission

SlotMachine

A PRIVACY-PRESERVING MARKETPLACE FOR SLOT MANAGEMENT

This Deliverable is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No 890456 under European Union's Horizon 2020 research and innovation programme.

Abstract

This document provides the requirements specification related to the SESAR ER4 project *SlotMachine*. It is recommended to read the upcoming deliverable *D2.3 Business Concepts* first, before looking into this deliverable *D2.1 Requirements Specification*. After that it makes sense to read *D2.2 System Design Document*.

SlotMachine employs blockchain technology and secure multi-party computation to extend the existing User-Driven Prioritisation Process (UDPP) solution with the possibility to keep private the participating airlines' confidential information such as the cost structure of flights. Technology will allow for secure, auditable transactions without the need for a central broker, whereby stakeholders will be able to enter slot swapping transactions without disclosing information to other participants. By demonstrating the feasibility of a privacy-preserving platform for swapping ATFM slots, the foundation can be laid for the development of a product that will be an essential element in the aviation industry in the future. It contributes to better use of existing resources at airports, higher efficiency of airlines, lower emissions, and shorter delays for passengers.

Table of Contents

List of Figures


SlotMachine


SESAR
JOINT UNDERTAKING


EUROPEAN UNION
EUROCONTROL

1 Introduction

1.1 Purpose of the document

The purpose of this document is to define a set of system requirements which reflect the operational requirements described in the upcoming D2.3 Business Concepts [1], which should be read before this deliverable).

The requirements included in this document follow the operational concept developed by PJ07-W2-S39 and are allocated to Phase 2 – "UDPP" [2].

This document covers functional, non-functional and interface requirements. They are addressing the "WHAT", not the "HOW", therefore they do not aim at specifying the physical design of the functional block (which is assigned to the industry), but the functional description and the necessary logical interfaces with other functional blocks.

1.2 Intended readership

The target group for this document is the SlotMachine project team with a special focus on R&D related topics. It provides input for the Advisory Board to describe detailed specification aspects and beyond the project, product managers and system architects can use the content of this document as a basis for implementation and further development.

1.3 Background

This document took references from the SESAR OSED (Operational Service and Environment Description) template structure and provides a comprehensive overview of the technical requirements. Those requirements are based on the business aspects described in D2.3 Business Concepts [1].

1.4 Structure of the document and relation to other deliverables

The document identifies the technical requirements for a SlotMachine system that could be used in an operational context. Those requirements are derived from the operational and business methods/use cases identified in D2.3 Business Concepts [1]. The technical requirements in this document are also the basis for the system design in D2.2. The remainder of this document is organized as follows. Chapter 2 contains the identified functional and non-functional requirements for the SlotMachine system. Chapter 3 describes iterative development plan for the project and structures functional components into logical groups for stepwise implementation and verification. Chapter 4 concludes the document. provides a short summary of this document and references relevant further documents.

2 Requirements

In this section we identify the functional and non-functional requirements that an operational SlotMachine system would have to fulfil. We primarily derive those requirements from the operational and business methods/use-cases identified in D2.3 Business Concepts [1], which align with the needs of airlines and network management. To this end, representatives of SWISS and EUROCONTROL reviewed the requirements of this document with respect to whether the technical requirements reflect their organizations' respective requirements. The first advisory board meeting in April 2021, which brought together different stakeholders and experts, provided additional insights that are also reflected in this document. In the following, we first give a brief, high-level overview of the envisioned SlotMachine system, its main components, and how those components work together before describing the identified functional and non-functional requirements for each of those components. For the different requirements we describe the rationale, particularly how each requirement relates to the operational requirements in D2.3 [1].

2.1 Overview

Figure 1 illustrates the main components of the SlotMachine system.

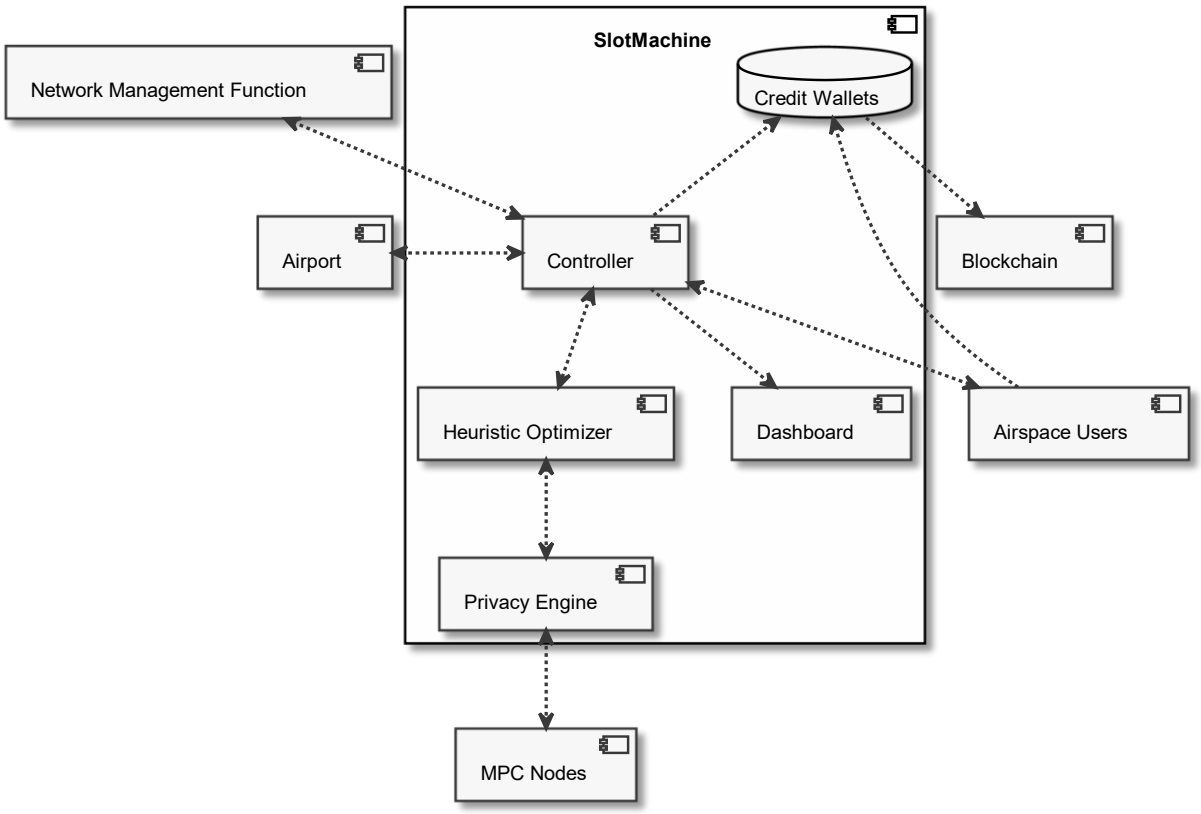

Figure 1. Main components involved in the flight prioritization process

Each requirement is assigned to one or more of the following stakeholders, which are described in detail in Deliverable 2.3 (Business Concepts):

- au – airspace users
- nmf – network management function
- apoc – airport operation centre
- mpo – marketplace operator

Requirements can include placeholder variables in ALL_CAPS. The actual values for such variables will be defined during prototype development and evaluation for selected scenarios. A list of all variables is provided in Section 2.4.

2.1.1 Component Descriptions

The **SlotMachine** system consists of multiple components, which we briefly describe in the following.

The **Controller** is the central component of the SlotMachine system, relaying the inputs from Network Management Function and Airspace Users to the Heuristic Optimizer to initiate a flight prioritization session for optimizing the flight prioritization in a given time window at an airport.

The **Heuristic Optimizer** employs an evolutionary optimization algorithm, e.g., a genetic algorithm, to find incrementally improved solutions to the flight prioritization problem in multiple iterations. After each iteration step, the Heuristic Optimizer sends the found solutions to the Privacy Engine for evaluation, which returns a fitness value computed over the encrypted preferences for the flights submitted by the airlines.

The **Privacy Engine** is responsible for the protection of sensitive data provided by airspace users as input to the optimization of flight prioritizations. At its core, it applies cryptographic techniques from the domain of multiparty computation—enabling computation on encrypted data—to assist the Heuristic Optimizer in finding flight prioritization solutions based on AU preferences. To do so, it manages a set of **MPC Nodes** which are carrying out the computation and provides an easy to use and secure interface for the heuristic optimizer to interact with.

The **Credit Wallets** record the credits amassed by the airlines over the course of the flight prioritization, which can be spent in optimization sessions in order to prioritize flights. A **Blockchain** may be used to make the transactions transparent to the stakeholders. The SlotMachine system will also have a **Dashboard** that shows various key performance indicators related to the flight prioritizations made via the system.

The **Airspace Users** have their own local user interface for communicating with SlotMachine. The user interfaces shall allow airlines to input preferences for flights based on margins, which are then automatically converted into a weight map, with sensitive values automatically encrypted.

The **Network Management Function** shall provide an initial flight prioritization and receives optimized flight prioritizations from the SlotMachine for confirmation.

The **Airports**, similar to the Airspace Users, have their own local user interface for communicating with SlotMachine. The user interfaces shall allow airports to identify slots the airport wishes to swap.

2.1.2 Sequence Diagram

The sequence diagram in Figure 2 illustrates the interaction between the different components of the SlotMachine system during an optimization session. The sequence diagram also includes a high-level view on the data flows. Data that do not have to be kept secret from the SlotMachine system are marked by «plain-text», sensitive data that are kept secret from the SlotMachine system and only processed in privacy-preserving manner through the multiparty computation (MPC) nodes are marked by «encrypted».

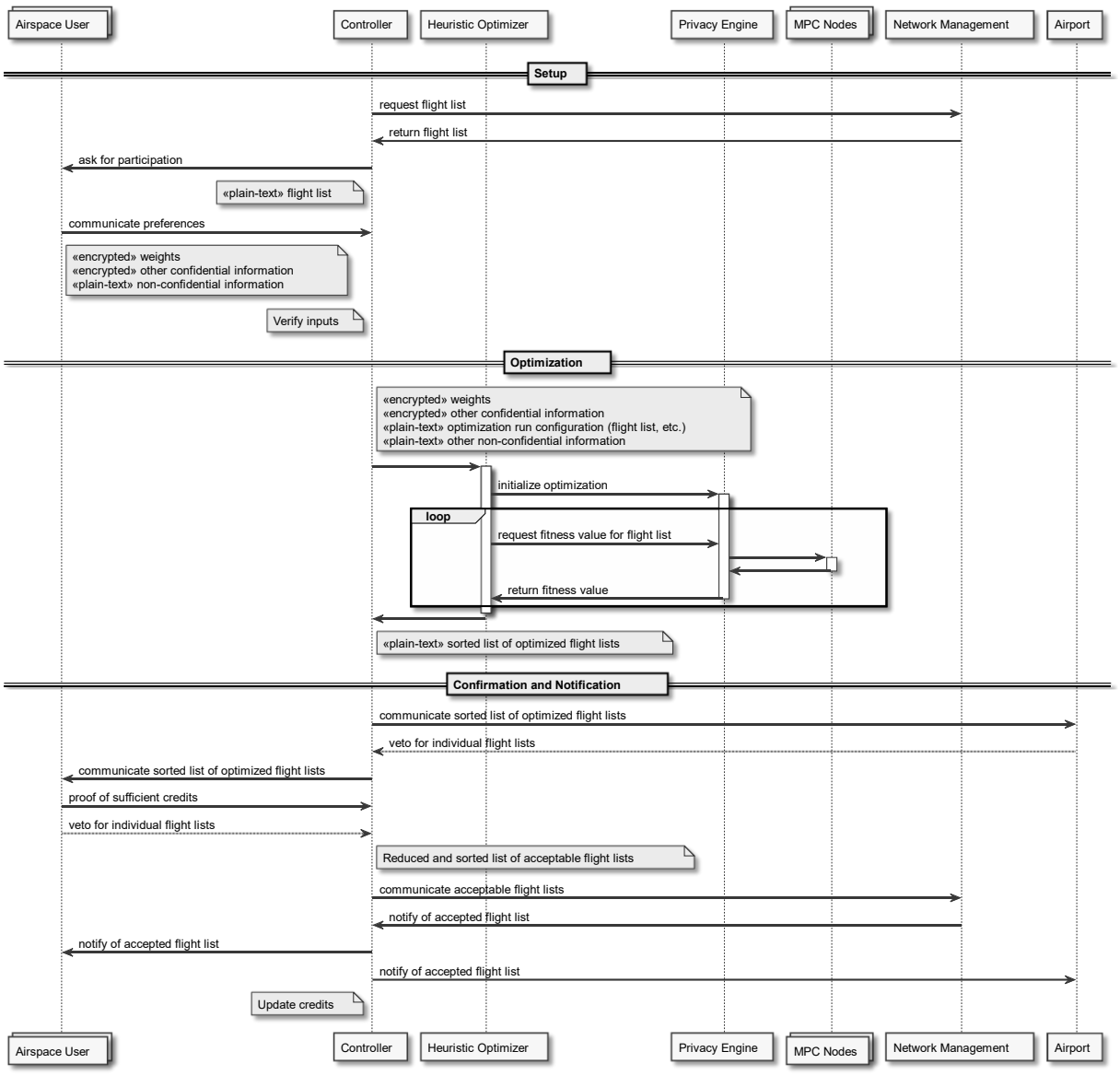

Figure 2. Sequence diagram illustrating the interaction and data flows between the components

In the setup phase, the Controller requests a flight list from the Network Management Function and dispatches the flights that can participate in an optimization session to the individual airspace users (AUs). The AUs then submit which flights they would like to enter in the optimization session along with the preferences for slots and possibly other encrypted and plain-text information. The Controller verifies the inputs from the AUs and initiates the optimization session by sending the collected information to the Heuristic Optimizer, which in turn initializes the Privacy Engine. The Heuristic Optimizer iteratively searches for optimal flight lists, which are evaluated by the Privacy Engine using MPC in order to keep the private inputs hidden even from the SlotMachine system itself. The Heuristic Optimizer finally returns one or more optimized flight lists to the Controller. The Controller communicates the optimized flight lists to the airport and the airspace users for confirmation. Finally, network management is informed and may or may not adopt the proposed new flight list. Finally, the airspace users and the airport are informed. Afterwards, the credits of the airspace users are updated according to the market mechanism (see D2.3) in order to compensate airspace users that give up a better slot for a flight.

2.2 Non-Functional Requirements

In the following we identify non-functional requirements related to (i) performance and scalability, (ii) portability and compatibility, (iii) reliability, availability, and maintainability, (iv) localization, (v) usability, (vi) privacy and security, and (vii) fairness. We use placeholders instead of specific values because at this stage of exploratory research we do not want to constrain development. Rather, these requirements should be seen as a guideline for future development of a product and the outcome of the SlotMachine project may provide hints as to what are realistic and sensible values for those placeholders.

2.2.1 Performance and Scalability

This section comprises REQs that describe throughput under a given workload for a specific time frame in each setting. Most of these requirements are derived from the time limits set due to the operational slot swapping process (see D2.3 Business Concepts [1]).

ID	Tags	Description
perf_1	au	The marketplace shall handle at least MIN_CONCURRENT_AIRLINES concurrent airlines. *Derived from the time limits set due to the operational slot swapping process (see D2.3 Business Concepts [1]).*
perf_2	mop	The solution shall support handling at least MIN_FLIGHT_NUMBER flights per flight prioritization session. *Necessary* requirements *as you need at least two flights to swap.*
perf_3	nmf	The found flight prioritization solutions shall be MIN_EFFICIENCY % more efficient than the existing flight list provided by the Network Manager in terms of the preferences submitted by the airspace users.

ID	Tags	Description
		The SlotMachine system should provide a gain with respect to the existing systems in place, otherwise it would not be needed. The gain can be measured by comparing the overall utility of the baseline flight list with the flight list found by the SlotMachine using the preferences (utilities of slots) submitted by the airspace users.
perf_4	au	A flight prioritization session should last at most MAX_SESSION_DURATION minutes. *Necessary REQ due to the synchronization process with the airport/airspace users, which takes some time.*
perf_5	au	Airlines and Airports shall be able to join with low effort in regard to usability and provided data. *Derived from the feedback from potential SlotMachine users (also stated within the first advisory board meeting). Stakeholders have to actually use the system in order to realize operational gains.*
perf_6	mop	The Heuristic Optimizer shall finish the optimization after reaching a specified threshold in terms of fitness/utility of the found solution or after a specified amount of time has passed for the optimization session and return the best flight prioritization found until that point. *See REQ perf_4. The optimization process cannot run forever and must return a result. The optimization algorithm must be chosen such that it will return a result no matter when it is aborted.*

2.2.2 Portability and Compatibility

Requirements to make sure that the system can be operated now and in the foreseeable future on the available platform infrastructure and also works together with adjacent systems. On the one hand, those REQs are derived from technical needs related to the state of the art of privacy-preserving computation and evolutionary optimization algorithms *(see [3] & [4])*. On the other hand, some REQs reflect industry best practices in application development.

ID	Tags	Description
port_1	mop	The interfaces of the different components shall be clearly defined and documented so that other stakeholders can develop new components. (cf. System Wide Information Management).
port_2	mop	Deployment of relevant components shall be flexible to cover different deployment use cases (see D2.3 Business Concepts [1]).
port_3	mop	The system design should offer a scalable and flexible architecture.

2.2.3 Reliability, Availability, and Maintainability

Requirements describing the accessibility of the system to the users at a given point in time and how to quickly recover from any failures. On the one hand, those REQs are derived from technical needs related to the state of the art of privacy-preserving computation and evolutionary optimization algorithms (*see [3] & [4]).* On the other hand, some REQs reflect industry best practices in application development.

ID	Tags	Description
rel_1	mop	SlotMachine components should be easy to deploy and configure.
rel_2	mop	A user manual shall provide installation and configuration instructions.
rel_3	mop	All software components shall be documented.
rel_4	mop	Software components shall be tested for their basic functionality.
rel_5	mop	SlotMachine components shall be centrally monitored and managed (dashboard).
rel_6	au	Essential information shall be stored in immutable and redundant form (e.g., in a blockchain). *Stakeholders would like to verify that the process is indeed fair* (see D2.3 Business Concepts [1])*, which was also verified in the first advisory board meeting.*
rel_7	mop	Privacy Engine shall be configurable per flight prioritization session.
rel_8	mop	Dynamic configuration of the MPC system (number of nodes, selection of nodes) per flight prioritization session shall be supported.
rel_9	mop	Configuration of MPC spare nodes shall be possible.
rel_10	mop	The system shall perform input validation.

2.2.4 Localization

Specific requirements in line with the context of the target audience.

ID	Tags	Description
loc_1	au	User interface components shall support multiple languages. *Derived by the need of operators to work with the UI in the most understandable language.*
loc_2	au	The user interface shall be available at least in English and German. *Due to the limited scope of the project, we focus on two languages only.*

2.2.5 Usability

Requirements that describe throughput under a given workload for a specific time frame in each setting.

ID	Tags	Description
usab_1	au	Any user interface towards airlines shall be made available as web application on a standard monitor. *Note: The project will not target mobile devices with small screen.* *In a first step, it is reasonable to assume that only a standard computer monitor shall be supported as the user interface.*
usab_2	au	An airspace user shall input into the user interface for each of its flights the preferences for slots in terms of "margins", i.e., the preferred time slot as well as the time not before and the time not after which the flight should not depart. From those margins, the user interface automatically derives a numeric utility value according to a specified valuation function. *Derived from the wallet/credits handling in the use cases definition (see D2.3 Business Concepts [1])*
usab_3	au	The user interface for airlines shall *automatically* encrypt protected information that must be hidden from the SlotMachine system and other actors. *Derived from the use case definition for end-to-end encryption (*see D2.3 Business Concepts [1]) and wished by AUs to be taken over by the user interface to lower the human workload.
usab_4	au	The user interface for airlines should be able to display improved flight prioritization results *Derived from the flight handling in the use cases definition (see D2.3 Business Concepts [1])*
usab_5	au	The user interface for airlines shall display information about improved flight prioritization results based on flight prioritization results from other airlines. *Derived from the flight handling in the use cases definition (see D2.3 Business Concepts [1]).*
usab_6	au	The Airspace User Web Interface should provide data capture mechanisms for flights to participate in the SlotMachine optimization. Only flights retrieved from the NMF shall be available for data capture. *Derived from the flight handling due to use cases definition* (see D2.3 Business Concepts [1]).
usab_7	au	The Airspace User Web Interface and the Airspace User Application Interface (REST endpoint) shall provide the same data capture and processing functionalities.

		Derived from technological need by using/extending already existing AUs web applications instead of using default AUs app.

2.2.6 Privacy and Security

REQs about privacy and security needs from the different stakeholders. These REQs are *derived from the use case definition for end-to-end encryption (see D2.3 Business Concepts [1]).*

ID	Tags	Description
priv_1	au	The data provided by the AU and identified as sensitive shall remain protected.
priv_2	au	The AU flight prioritization preferences shall remain confidential and secured from competitors.
priv_3	au	The AU flight prioritization preferences shall remain confidential and protected from honest-but-curious platform operator individuals.
priv_3.1	au	The AU flight prioritization preferences shall remain confidential and protected from Network Management Functions (incl. Flow Management Positions).
priv_4	au	The AU flight prioritization preferences shall be processed in encrypted/encoded form only in the platform.
priv_5	au	The internally used representation of AU flight prioritization by SlotMachine shall be securely derived from AU input for each flight prioritization and flight considered (e.g. weights).
priv_6	au	Incorrect input data from AUs shall be detectable, e.g., invalid margins and weight configurations.
priv_7	au	All interactions between AU and the platform must be authentic and secured, i.e., authentication and authorization mechanisms must be in place to identify users/roles and their privileges.
priv_8	au	Security shall be maintained against honest-but-curious behaviour of Privacy Engine service operators.
priv_9	au	Security shall be maintained against honest-but-curious behaviour MPC nodes.
priv_10	au	Security shall be maintained against malicious behaviour of AUs.
priv_11	au	The credits allocated by each airline to its flights must remain private.
priv_12	au	The global balance of the credit must be published to all participants after a given time (CREDIT_PUBLISHING_INTERVAL) to ensure transparency and equity.
priv_13	au	The credit balance of a participant must be accessible for the respective participant at any time.

ID	Tags	Description
priv_14	au	The credit balances must be consistent and immutable for any colluding minority in the system, including the platform operator.
priv_15	au	The executed flight prioritization should be transparent. *This is due to the fact that the final sequence is public anyway (checked with NMF and can be observed on runway). The individual AU inputs, however, are not revealed and zero-knowledge proof techniques will be used to show that they were correct.*
priv_16	au	The final executed flight prioritization should be accessible for all AUs.
priv_17	au	The history of flight prioritization should be stored in a consistent and immutable form.

2.2.7 Fairness

REQs describing the balance between airspace users regarding flight prioritization. These REQs are derived from the desired features of a market mechanism and the business use cases (see D2.3 Business Concepts [1]), but also partially verified during first advisory board meeting.

ID	Tags	Description
fair_1	au	The system optimisation models must be transparent to the users.
fair_2	au	The system calculations concept leading to the decision of the solution to implement must be transparent to the users.
fair_3	au	The flexible credits to be used cannot be related in any matter to financial means.
fair_4	au	No other stakeholders than the airlines operating flights caught in the given regulation can participate.

2.3 Functional Requirements

This section lists relevant requirements grouped by the components described in Figure 1. The sequence diagram in Figure 2 illustrates the interaction and data flows between the components.

2.3.1 SlotMachine

The SlotMachine system comprises the components for optimizing the flight prioritization at an airport in a privacy-preserving way and controls the flight prioritization. The SlotMachine also maintains a database of credits awarded to airspace users. Airspace users shall interact with the SlotMachine using the provided Web User Interface or the application interface (REST endpoint). The SlotMachine communicates the optimized flight prioritization to the network management function, airport, and airspace users.

ID	Tags	Description
sm_1	au	An Airspace User shall communicate preferences to the SlotMachine system. *Derived from the flight handling in the use cases definition (see Business Concepts).*
sm_2	au	The SlotMachine system shall maintain wallets for managing credits of Airspace Users. *Derived from the wallet/credits handling in the use cases definition (see Business Concepts [1]).*
sm_3	au	The SlotMachine system shall provide the AUs with access to their wallet to check upon their credit status. *Derived from the wallet/credits handling in the use cases definition (see Business Concepts [1]).*
sm_4	au	A distributed ledger (blockchain) shall be used by the SlotMachine to document credit transactions. *Derived from the wallet/credits handling in the use cases definition* (see *D2.3 Business Concepts [1]).*

2.3.2 Controller

The Controller is the central component of the SlotMachine, relaying messages between Airspace User, Network Management Function, Heuristic Optimizer, Airport, and Credit Wallet, which includes functions of the controller in relation to the credit wallets. These REQs are derived from the desired features of a market mechanism and the business use cases (see D2.3 Business Concepts [1]).

ID	Tags	Description
co_1	mop	The Controller shall receive the preferences communicated by the AU.
co_2	mop	The Controller shall retrieve the current flight list from the Network Management Function (NMF).
co_3	nmf	The Controller shall communicate a ranked list of optimized flight prioritizations to the NMF.

co_4	mop	The Controller shall call the Heuristic Optimizer to receive a ranked list of optimized flight prioritization based on the initial flight prioritization received from the NMF.
co_5	apoc	The Controller shall ask stakeholders (e.g., airport, AUs) for confirmation of the proposed flight prioritization.
co_6	au	The Controller shall ask participating AUs for confirmation of the proposed flight prioritizations and offer AUs the possibility to discard unacceptable solutions before submission to NMF.
co_7	au	The Controller shall provide a method for providing actual credit amount for AUs stored in the Blockchain.
co_8	au	The Controller shall provide mechanism for updating AUs credits stored in the Blockchain.

2.3.3 Heuristic Optimizer

The Heuristic Optimizer employs an evolutionary optimization algorithm to find an optimal flight prioritization given the preferences submitted by the AUs. These requirements are derived in parts from technical characteristics of multiparty computation and evolutionary algorithms, which are described in the analyses of the state of the art *(see [3] & [4]).*

ID	Tags	Description
ho_1	mop	The Heuristic Optimizer shall evaluate each flight prioritization independently from other flight prioritizations.
ho_2	mop	The Heuristic Optimizer shall receive information about flights from the Controller.
ho_3	au, mop	The Heuristic Optimizer shall receive encrypted flight prioritization preferences of AUs from the Controller.
ho_4	mop	The Heuristic Optimizer shall receive public flight information from the Controller.
ho_5	mop	The Heuristic Optimizer shall generate flight prioritizations under consideration of public flight information.
ho_6	mop	The Heuristic Optimizer shall initialize a Privacy Engine (Privacy Engine) session with encrypted flight prioritization preferences.
ho_7	mop	The Heuristic Optimizer shall use Privacy Engine to evaluate fitness of generated flight prioritizations.
ho_8	mop	The Heuristic Optimizer shall return a ranked list of flight prioritizations to the Controller.
ho_9	mop	The Heuristic Optimizer shall always return a solution, independent of the run time, meaning that an optimization can be aborted at any time and still return a valid result.

		See REQ perf_6.

2.3.4 Privacy Engine

The Privacy Engine is managing private inputs from AUs in encrypted form, such that computations on the data is still possible. The main responsibility is to assist the Heuristic Optimizer to do its work in a privacy preserving way, i.e., by not revealing AU inputs. These requirements are derived from technical needs identified in the state of the art *(see [3] & [4]).*

ID	Tags	Description
pe_1	mop	The Privacy Engine shall provide the functionality to assess the fitness of provided flight prioritizations.
pe_2	mop	Privacy Engine shall act in a privacy preserving manner (sensitive information is never decoded).
pe_3	mop	Privacy Engine shall not persist session data.
pe_4	mop	Privacy Engine shall provide an interface to be used by the optimization component for querying fitness of flight prioritization exchange solutions.
pe_5	mop	Privacy Engine shall only communicate with the heuristic optimiser.
pe_6	mop	Privacy Engine shall manage flight prioritization exchange sessions to preserve namespaces for dedicated optimization instances.
pe_7	mop	Privacy Engine shall calculate and return fitness values for a given set of flight prioritization exchanges to be considered per iteration.
pe_8	mop	Privacy Engine shall be configurable to support different fitness functions.
pe_9	mop	Privacy Engine shall provide means to compute anonymized statistics and KPIs.
pe_10	mop	Privacy Engine shall manage the usage of MPC nodes used for processing of encrypted data.
pe_11	mop	Privacy Engine shall monitor and manage the network of MPC nodes.
pe_12	mop	Privacy Engine shall assure that sensitive data is only passed in encrypted form and never available in plaintext within PE.
pe_13	mop	Privacy Engine shall allow for sanity checking of AU inputs in a privacy preserving but verifiable manner. *Note: Current options are Zero Knowledge Proofs or performed within the MPC system*
pe_14	mop	Privacy Engine shall support public verification means for selected key indicators.
pe_15	mop	Privacy Engine shall provide a dedicated pre-processing service for encoding and encrypting input data provided by the AU.

ID	Tags	Description
pe_16	mop	Privacy Engine pre-processing service shall be deployable at AU client side for local post-processing of sensitive input data.
pe_17	mop	Privacy Engine shall be made robust against malicious behaviour of Privacy Engine pre-processing component (malicious AU model).

2.3.5 MPC Nodes

MPC (Multiparty Computation) nodes are the basic compute elements used by Privacy Engine to act on encrypted data. Multiple MPC nodes are used in a distributed setting to jointly hold the encrypted input data from AUs. They are also able to jointly compute functions on the private inputs only revealing the results upon request by the Privacy Engine. These requirements are derived from technical needs identified in the state of the art *(see [3] & [4]).*

ID	Tags	Description
mpc_1	mop	MPC nodes shall be able to compute functions on secret shared (encrypted) data.
mpc_2	mop	Individual MPC nodes shall run as standalone service (dedicated process or container).
mpc_3	mop	MPC nodes shall be able to accept compute tasks from the Privacy Engine.
mpc_4	mop	MPC nodes shall not be accessible from other components than the Privacy Engine.
mpc_5	mop	MPC shall be able to decrypt sensitive input data from AUs which have been encoded by the Privacy Engine pre-processing service in a privacy preserving way (secret shared).
mpc_6	mop	MPC nodes shall be able to maintain sensitive data in secret shared form only.
mpc_7	mop	AU input data shall only be valid for one specific flight prioritization.
mpc_8	mop	AU sensitive data shall not be stored beyond the session they are used in and securely deleted.
mpc_9	mop	MPC shall only execute agreed and known functions.
mpc_10	mop	External components shall not be able to readout or extract confidential information beyond executing specified functions.
mpc_11	mop	MPC nodes shall be able to jointly compute predefined functions (fitness values, statistics) for the selected flight prioritization.
mpc_12	mop	MPC nodes shall be able to securely communicate with each other during function evaluation
mpc_13	mop	Accepted tasks from Privacy Engine shall be jointly executed in a pre-configured compound of nodes.

ID	Tags	Description
mpc_14	mop	The compound of nodes to be used shall be selected on session basis.
mpc_15	mop	MPC nodes shall be able to check inputs for consistency if necessary.
mpc_16	mop	MPC nodes shall be able to check inputs against public information available in ledgers (e.g. commitments) if necessary.
mpc_17	mop	MPC nodes shall be able to generate proofs for public verification of selected results if necessary.

2.3.6 Airspace User

The Airspace User component is the user interface that runs locally at the airlines and airports to manage Flight and swop results.

ID	Tags	Description
au_1	au	The Airspace User (AU) shall have a user interface for entering times (NotBefore, NotAfter, TimeWished, priority) or alternatively directly the weight map for the flight list. *Derived from the flight handling in the use cases definition (see D2.3 Business Concepts [1]).*
au_2	au	AU shall receive a list of flights that can be swoped and available slots from the Controller. *Derived from the flight handling due to use cases definition (see Business Concepts [1]).*
au_3	au	AU shall submit an encrypted weight map for each flight detailing that flight's preferences regarding the available flight prioritization. *Derived from the use case definition for end-to-end encryption (see Business Concepts [1]).*
au_4	au	AU shall submit an encrypted map for credits willing to spend for each flight in a flight prioritization. *Derived from the use case definition for end-to-end encryption (see Business Concepts [1]).*
au_5	au	AU shall authenticate themself before using AU App. *Derived from the deployment use cases (see D2.3* Business Concepts [1]*), where different stakeholders need to be authenticated and authorized.*
au_6	au	AU App should be able to inform AU about flight optimization results on an active base and provide the possibility to accept or discard the flight optimization.

		Derived from the flight handling in the use cases definition (see D2.3 Business Concepts [1])
au_7	au	AU App shall provide the information when the next optimisation is started. *Derived from the use case definition for optimisation routine (see D2.3 Business Concepts [1]).*

2.3.7 Network Management Function

The Network Management Function (NMF) component supplies the SlotMachine with an initial flight prioritization for an optimization session. SlotMachine then returns a ranked list of flight prioritizations to the NMF for confirmation.

ID	Tags	Description
nm_1	nmf	Network Management Function (NMF) shall provide an initial flight prioritization. *Derived from the flight handling in the use cases definition (see D2.3 Business Concepts [1])*
nm_2	nmf	NMF shall confirm which of the proposed flight prioritizations was selected. *Derived from the flight handling in the use cases definition (see D2.3 Business Concepts [1]).*

2.3.8 SlotMachine Dashboard

Key performance indicators (KPIs) shown in a dashboard shall document the success of the SlotMachine system.

ID	Tags	Description
db_1	mop	The SlotMachine Dashboard shall aggregate metadata and KPIs from flight prioritization exchanges to allow statistical evaluation of the SlotMachine performance. *Derived from display of aggregated insights and analytics in the use cases definition (see D2.3 Business Concepts [1])*

2.4 Requirement Placeholder

This section lists all requirement variables used in the non-functional and functional requirements:

Variable	Description
CREDIT_PUBLISHING_INTERVAL	interval between publishing current credits of all AU participants
MAX_SESSION_DURATION	maximum duration in minutes after triggering a new flight prioritization exchange session
MIN_CONCURRENT_AIRLINES	minimum number of supported concurrent airlines
MIN_EFFICIENCY	minimum efficiency gain compared to initial flight prioritization
MIN_FLIGHT_NUMBER	minimum number of flights per flight prioritization exchange session

3 Development Roadmap

Based on requirements described in Chapter 2, the following prototype options are considered for implementation during the development phase. To cover a broad range of options while still maintaining agility in development, we consider different deployment scenarios for SlotMachine and will incrementally implement various prototypes covering a subset of the requirements.

3.1 Deployment Scenarios

The following deployment scenarios are considered based on the use cases described in D2.3 Business Concepts, Section 2.4.

- Deployment Use Case – Scenario A: Centralized Environment: Centralized deployment hosted by Network Manager: all server components are hosted by a trusted party (here: Network Manager) and AUs connect via local clients.

- Deployment Use Case – Scenario B: Hybrid Environment: Centralized instances at airport: SlotMachine server components are hosted at an airport and AUs having flights departing from this airport register to this SlotMachine instance; multiple SlotMachine instances from various airports synchronize with the Network Manager by submitting flight prioritization exchange proposals.

- Deployment Use Case – Scenario C: Decentralized Environment: Decentralized solution among participating AUs: a decentralized SlotMachine hosted by AUs which run their own MPC nodes and synchronizing with airports and the Network Manager.

3.2 Economic Scenarios

Different scenarios for a market mechanism for compensation of AUs giving up slots are considered in D2.3 Business Concepts [1]. Accordingly, the feasibility of the following market mechanisms will be evaluated using the prototypes:

- AUs specify preferences for slots in terms of utility and receive or pay
    - Cash money based on the utility.
    - Credits based on the utility.
- AUs indicate willingness to move certain flights backwards in exchange for credits or willingness to pay credits to move a certain flight forward.

3.3 Prototype Iterations

During the lifetime of the project a prototype to demonstrate, evaluate, and compare certain functions will be developed according to D2.3 Implementation Use Case" using an iterative approach, which means functionalities and components will be added step by step. The identified requirements will guide the implementation of different prototypes which are developed iteratively. Ideally the final iteration should satisfy all identified requirements. However, part of the project is to find out whether the requirements can be satisfied by SlotMachine implementation or whether they are even necessary. In particular, we will develop the following iterations of the prototype:

1. The first version of this prototype "**non-privacy-preserving**" will cover initial demonstration of submitting preferences by AUs (via the AUs WebClient through the Controller to the Optimizer) and optimizing a flight prioritization to discuss first results. This prototype involves basic implementation of following components:
    - AUs WebClient
    - Controller
    - NMF
    - Optimizer
    - Dashboard
2. In the 2nd iteration of the **prototype privacy-preserving** mechanisms will be added through the Privacy Engine (incl. MPC nodes) and end-to-end encryption between AU clients and MPC nodes; a Dashboards provide overview and aggregated insights over the optimizations performed by the components. This prototype involves basic implementation of following components:
    - AUs WebClient
    - Controller
    - NMF
    - Heuristic Optimizer
    - MPC Nodes
    - Dashboard
3. In the final version, **secure, and traceable credit handling** will be provided through blockchain technology to allow greater flexibility and more options for airlines participating in the marketplace. In this state all mentioned Components will be involved.

4 Conclusions

This document comprises technical requirements for the SlotMachine system based on the business and operational methods and use cases described in D2.3 Business Concepts [1]. Additionally, this document defines which concrete prototype will be implemented in three steps, each step realizing different subsets of those requirements.

In the upcoming D2.2 System Design Document, the design of the SlotMachine system is elaborated further based on the inputs from this document.

5 References

[1] SESAR: “Business Concepts”. D2.3, SlotMachine, 890456. 28 July 2021 un-official draft.

[2] SESAR: “Final OSED/SPR”. D3.1.001, PJ.07-W2-39-V3. 28 November 2020.

[3] SESAR: “Report on State-of-the-Art of Relevant Concepts”. D3.1, SlotMachine, 890456. 28 July 2021.

[4] SESAR: “Report on State-of-the-Art of Relevant Concepts”. D4.1, SlotMachine, 890456. 30 July 2021.

Appendix A Terms of Glossary and Abbreviations

Term	Definition
Airport slot	Permission to operate a service at a given time (with regard to flight arrivals and departures).
Airspace User (AU)	Airlines and low volume users (LVUC) including Business aviation.
Air traffic flow management (ATFM) slot	ATFM departure slot, assigned tactically by the Network Manager to manage congestion.[1]
Baseline Delay	Amount of delay a flight or group of flights would be assigned if no UDPP prioritisation were applied.
Cancellation	A flight is cancelled if it is not operated (usually reallocating the passengers to other flights). See also 'suspension'.
Capacity	Maximum number of flights that can enter into a sector or airport per unit of time (usually 1 hour, but can be defined for any time-window length, e.g., 15 minutes).
Capacity Constrained Situation	A period of capacity and demand imbalance in which the capacity reduction and the resulting excess demand causes stress in the ATFM slot allocation process, relative to that allocated previous to the imbalance.
Capacity surplus	Difference between capacity and demand when the available capacity is enough to allocate the actual demand for a given period of time and there is still room to potentially allocate a higher number of flights.
Cost of delay	Economic cost incurred by an AU due to the delay experienced by a flight.
Credit	For ease of reading, in this document, the term 'credit' will often be used as a substitute for 'Delay Credit'.
Delay	The difference between the ATFM slot and the scheduled time of departure.
Delay Credit	Unit of the virtual currency (i.e. credits) as an expression of the value generated to other users (positive externalities) or the loss caused (negative externalities) in terms of delay.

[1] https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=LEGISSUM:tr0032&from=EN

Term	Definition
Demand	Total number of flights that plan to enter into a sector or airport in a given unit of time (usually defined for 1 hour but can be also defined for any time-window length, e.g. 15 minutes).
Equity	Equity, measures how uniformly the distribution of the good is performed, that is, without taking into account individual satisfaction thresholds.
Excess demand	Difference between demand and capacity when the available capacity is not enough to allocate the demand for a given period of time.
Externalities	Collateral effects caused by the decision/action of an AU regarding the usage (or non-usage) of a certain slot. When such decisions change (limiting or expanding), it allows other AU to use the same or other slots.
Fairness	Fairness can be defined as the quality of distributing something among a set of individuals in a manner such that each receives a share that fulfils its individual satisfaction threshold. In order to measure fairness objectively, it is essential to agree on a common way to quantify such individual satisfaction thresholds.
Flight list	A sequence of flights.
Flight prioritization session	A group of departing aircrafts in a certain time span that is optimized for their starting sequence based on AU inputs.
Flight prioritization (solution)	Best possible solution found by SlotMachine within a flight prioritization exchange session.
Hotspot	Similar to CCS, but while CSS refers to periods of demand and capacity imbalance at airports, HSPT refers to periods of demand and capacity imbalance in en-route sectors.
Multiparty Computation (MPC)	Secure multiparty computation (MPC / SMPC) is a cryptographic protocol that distributes a computation across multiple parties where no individual party can see the other parties' data.[2]
Near on-time	Delay higher than 0' but ≤ 15' for an AU.

[2]https://en.wikipedia.org/wiki/Secure_multi-party_computation

Term	Definition
Network Management Function (NMF)	The Network Management Function is the role that is currently filled by the Network Manager.
Network Manager	The body entrusted with the tasks necessary for the execution of the functions referred to in Article 6 of Regulation (EC) No 551/2004.[3]
On-time	Delay = 0' for an AU.
Operational Service and Environment Description (OSED)	The Operational Service and Environment Definition (OSED) document describes the operational concept defined in the Detailed Operational Description (DOD) in the scope of its Operational Focus Area (OFA). It defines the operational services, their environment, scenarios and use cases and requirements.[4]
Protection	A flight is protected if the AU sends a request (manually or through automated means) to allocate the flight to a particular slot, usually between the scheduled time (i.e., zero delay) and the slot assigned by FPFS (i.e., the Baseline Delay). The flight is *actually* protected only if no other flights with higher priority express the desire of using the same slot (i.e., protect their flights in the same slot).
Recovery period	Period of time in which there is a free slot where the last flight of the excess demand can be allocated and therefore the system is no longer under stress, i.e., the excess demand is fully absorbed.
RESTful	A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for representational state transfer and was created by computer scientist Roy Fielding.[5]
Slot	Period of time within which a given resource can be used by a specified user only (see also 'airport slot' and 'ATFM slot'). NB. By default, in this document, 'slot' refers to ATFM slot and assume slot widths of 1 minute.
Stress period	Period of time in which there is an imbalance between demand and capacity, and thus an excess demand is present.

[3] https://www.skybrary.aero/index.php/Network_Manager

[4] https://www.sesarju.eu/sites/default/files/documents/solution/Sol107%204%20Point%20Merge%20Complex%20TMA_OSED.pdf

[5] https://www.redhat.com/en/topics/api/what-is-a-rest-api#:~:text=choose%20Red%20Hat%3F-,Overview,by%20computer%20scientist%20Roy%20Fielding.

Term	Definition
Suspension	A flight is suspended if it is operated out of (after) the CCS period, i.e. from the recovery period onwards.
User-Driven Prioritisation Process (UDPP)	User-driven prioritisation process (UDPP) allows airspace users to minimise the impact of delay in capacity constraint situations. The UDPP process solves a capacity constraint by attributing ground delays based on airspace user preferences.[6]
Value of Suspension	Value in terms of credits that a flight generates to the network if suspended (or cancelled). It is calculated as the sum of all the positive externalities (i.e., minutes of delay reduced) generated for each individual flight involved in a CCS.

[6]https://www.eurocontrol.int/use-cases/implementing-user-driven-prioritisation-process-zurich-airport#:~:text=User%2Ddriven%20prioritisation%20process%20(UDPP,based%20on%20airspace%20user%20preferences.


SlotMachine


SESAR
JOINT UNDERTAKING


AIT
AUSTRIAN INSTITUTE OF TECHNOLOGY
TOMORROW TODAY


FREQUENTIS
FOR A SAFER WORLD


JKU
JOHANNES KEPLER
UNIVERSITÄT LINZ


SWISS