Report on testing of advanced performance data model

D3.3

AURORA

Grant:	699340
Call:	H2020-SESAR-2015-1
Topic:	Sesar-11-2015 ATM Performance
Consortium coordinator:	CRIDA A.I.E.
Edition date:	24th Apr 2018
Edition:	00.02.02

Founding Members
EUROPEAN UNION
EUROCONTROL
SESAR JOINT UNDERTAKING

Authoring & Approval

Authors of the document

Name/Beneficiary	Position/Title	Date
Aditya Grover / CeADAR	Project Member	28/01/2018
Brian Mac Namee / CeADAR	Project Member	28/01/2018
Shen Wang / CeADAR	Project Member	28/01/2018
L. D ‘Alto / BR&TE	Project Member	28/01/2018

Reviewers internal to the project

Name/Beneficiary	Position/Title	Date
Pablo Sánchez-Escalonilla / CRIDA	Project Coordinator	20/02/2018
Javier López-Leones / BR&TE	Project Member	20/02/2018
Philip Phantholt / FR24	Project Member	20/02/2018
Brian Mac Namee / CeADAR	Project Member	20/02/2018

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

Name/Beneficiary	Position/Title	Date
Pablo Sánchez-Escalonilla / CRIDA	Project Coordinator	08/03/2018
Javier López-Leones / BR&T	Project Member	08/03/2018
Brian Mac Namee / CeADAR	Project Member	08/03/2018
Philip Phantholt / FR24	Project Member	08/03/2018

Rejected By - Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date

Document History

Edition	Date	Status	Author	Justification
00.00.01	28/01/2018	Draft	B. Mac Namee	New Document
00.00.02	05/02/2018	Draft	S. Wang, A. Grover	Main Contents Added
00.00.03	09/02/2018	Draft	S. Wang	Minor Corrections
00.00.04	12/02/2018	Draft	B. Mac Namee	Full Review
00.00.05	16/02/2018	Draft	L. D ‘Alto	Section 2.3
00.00.06	26/02/2018	Draft	B. Mac Namee	Full Review
00.00.07	28/02/2018	Draft	B. Mac Namee	Revised executive summary and conclusions

				and added project description.
00.01.00	08/03/2018	Final version for submission	F. Celorrio	Consolidated for submission
00.01.01	20/04/2018	Draft revision	Shen Wang	Revised submission based on feedback from the commission
00.01.02	24/04/2018	Draft revision	Brian Mac Namee	Revised submission based on feedback from the commission
00.02.02	25/04/2018	Final version for submission	Pablo Sánchez-Escalonilla	Consolidated for submission

AURORA

ADVANCED USER-CENTRIC EFFICIENCY METRICS FOR AIR TRAFFIC PERFORMANCE ANALYTICS

This document is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No 699340 under European Union's Horizon 2020 research and innovation programme.

Abstract

This document describes and explains the methodology and results of testing the advanced performance data model developed. In particular, the overall motivation and evaluation plan and key results of proposed advanced performance data model are firstly introduced, followed by a detailed description of the stream-based data model testing results against a list of success criteria for each of the four objectives, proposed in the earlier deliverable. Finally, the conclusions of the full testing are represented along with the limitations when applied in production / realistic operation level.

Table of Contents

Executive Summary

This document describes the results of the experiments performed to verify the performance of the advanced performance data model designed for calculating online efficiency indicators in the AURORA project. The experimental plan described in AURORA deliverable 3.1 [2]was followed. The experimental plan specified four verification objectives:

- **VER-OBJ-ONLINE-2.1.1 Read and process data in a streaming context**

  It has been shown that large sample streams of surveillance data can be ingested into the online platform without significant problems in terms of duplicate and out of order signals.
- **VER-OBJ-ONLINE-2.1.2 Calculate the flight efficiency indicators accurately along the flight trajectory**

  The advanced performance data model has been shown to be capable of calculating a complete set of AURORA performance indicators in near real-time without significant divergence from the indicators calculated offline. The prediction of indicators was not completed as additional algorithms were necessary to generate new full trajectories that had to comply with the requirements to run the generation and reconstruction processes, thought the consistent integration of the already flown trajectories based on ADS-B with the predicted trajectories until destination.
- **VER-OBJ-ONLINE-2.1.3 Calculate indicators efficiently enough to match specifications of AURORA STAM use cases i.e. the method to calculate off-line (trajectories generation, reconstruction and optimal routes) in combination with advanced analytics is feasible to obtain the efficiency indicators online**

  AURORA airspace users' workshops defined an upper limit on the latency with which indicators should be updated after receiving a new surveillance point of 5 minutes and a throughput target of approximately 350 surveillance updates per second. The advanced performance data model has been verified to be capable of operating within these specifications.
- **VER-OBJ-ONLINE-2.1.4 Select the most appropriate streaming data technology for the AURORA STAM use case**

  A series of industry-standard big data streaming technologies have been selected and used to build the platform.

Although almost all verification objectives have been met successfully there are recommendations for future developments. The most important of these are as follows:

- The data cleaning process should be better integrated with the data pipeline.
- The live trajectory reconstruction and generation services were not used during the verification experiments as current implementations do not operate with low enough latency to support the rest of the advanced performance data model. Updating these services should not be a significant task, however, and they could be used in further experiments.

Currently, flight efficiency indicators can only be calculated after the completion of a flight. By using a data streaming technology and live flight data, however, the AURORA project has verified that indicators can be calculated online in near real time. This dynamic calculation will make efficiency indicators available to airspace users while flights are live and could be used to identify efficiency and equity problems during flights, to better plan STAM measures, and to monitor the performance of groups of flights.

1. Introduction

1.1 Purpose of the document

This document provides a report on testing the advanced performance data model proposed and implemented for AURORA [1]. The report describes the results of the experiments carried out to evaluate the feasibility and performance of implementing online calculation of the efficiency indicators developed in the AURORA project [2].

1.2 Intended readership

This document is intended to be used by AURORA members, by the SJU reviewers, and by the SESAR 2020 partners addressing the definition of the performance framework.

1.3 Acronyms and Terminology

Term	Definition
ADS-B	Automatic Dependent Surveillance-Broadcast
AIDL	Aircraft Intent Description Language
Apache Kafka	A distributed log messaging system from Apache open source software community
Apache Spark	An in-memory distributed big data processing system from Apache open source software community
API	Application Programming Interface
ATFCM	Air Traffic Flow and Capacity Management
ATM	Air Traffic Management
AU	Airspace User
AURORA	Advanced User-centric efficiency metRics for air traffic perfORmAance metric
BADA	Base of Aircraft Data
BRTE	Boeing Research & Technology Europe
CeADAR	Centre for Applied Data Analysis Research
CFMU	Central Flow Management Unit
CI	Cost Index
CPU	Central Processing Unit
DDR	Demand and Data Repository
DDR2	Demand and Data Repository 2
ECAC	European Civil Aviation Conference

Term	Definition
EU	European Union
FR24	FlightRadar 24
FREE_CI	Free routing trajectory optimizing the cost of flight time and fuel (Cost Index $\neq$ 0)
GEO_FP	Geodesic trajectory following the flight plan
GFS	Global Forecasting System
Horizon 2020	EU Research and Innovation programme implementing the Innovation Union, a Europe 2020 flagship initiative aimed at securing Europe's global competitiveness.
I/O	Input and Output
IEEE	Institute of Electrical and Electronics Engineers
INCEPT	Aircraft Intent Generation and Trajectory Synthesis Service
INTRAC	Aircraft Intent Inference and Trajectory Reconstruction service
INTRO	Intent-based Trajectory Optimization Service
IPOPT	Interior Point OPTimizer
JSON	File format of FR24 data
KEA	Key performance Environment indicator based on Actual trajectory
LAT	Latitude
LON	Longitude
OEW	Operating Empty Weight
OPT_CI	Trajectory following the flight plan that optimizes the cost of flight time, fuel and taxes
PostGIS	A geographic information system extension based on postgresSQL database.
QAR	Quick Access Recorder
RECON	Reconstructed trajectory using ADS-B data
SAMBA	A network protocol for sharing files with multiple users
SESAR	Single European Sky ATM Research Programme
SJU	SESAR Joint Undertaking (Agency of the European Commission)
STAM	Short-Term ATFCM Measures
UP	Reconstructed trajectory following the flight plan

Table 1 Acronyms and Terminology

1.4 Project introduction

AURORA responds to the first Call for Proposals of SESAR Exploratory Research projects launched under Part III `Societal Challenges´ of the Horizon 2020 Research Framework Programme (H2020-

SESAR-2015-1). AURORA addresses the Research Topic 11-2015: *ATM Performance* and in particular the need to explore promising new performance indicators for operational efficiency based on aircraft operators' needs.

AURORA will propose new metrics to assess the operational efficiency of the ATM system. These new metrics will be developed with the aim of encapsulating the airspace users' operational objectives, considering fuel consumption, schedule adherence and cost efficiency of the flights. User-preferred trajectories will be defined as the reference for performance analysis purposes. AURORA will also propose metrics to measure how fairly the inefficiencies in the system are distributed among the different airspace users. These metrics will serve to quantify the differences in the inefficiencies experienced by the different airspace users in a given operational context. Moreover, the metrics will be aligned with the Performance Scheme trying to achieve the performance objectives proposed by SESAR2020.

One of the key elements to obtain the new efficiency indicators is the calculation of an optimal or baseline reference representing the achievable target efficiency that the airspace user assigns to any given flight. In other words, the definition of the optimal or user-preferred trajectory (i.e. the business trajectory in SESAR) will play a key role in the definition and assessment of the new indicators. AURORA will develop a method to model user-preferred trajectories by combining cost and trajectory models without requiring confidential airspace users' information.

The other main research area proposed by AURORA will consist of exploring and testing techniques borrowed from the data science and information management fields for the collection and aggregation of data. These techniques will allow AURORA to propose a new framework for ATM decision-making based on real-time performance monitoring. In this new framework, the ATM decision-making processes will be supported by live indicators of actual operational performance and realistic achievable targets, where the airspace users could take an active role.

1.5 Document structure

The document is structured in the following sections:

- **Section 1**: This section introduces this document, including the objectives, intended readership, the organisation of the document, and the list of acronyms and terminology used.
- **Section 2**: This section contains a comprehensive overview of the key findings in this report. In particular, it includes an overview of the tested stream based data model with its limitations and assumptions, a list of calculated flight efficiency indicators, an introduction to the testing plan and traffic scenario, key results against each success criteria, and deviations from the planned activities
- **Section 3**: This section describes the detailed results for "*Exercise 2-1 Verification of the methods for the on-line calculation of indicators*". In this section, the success criteria are assessed for each verification objective depending on the results obtained during the experiments.
- **Section 4**: This section includes conclusions, and recommendations in terms of technical feasibility and operational benefits.
- **Section 5**: This section outlines the references which are cited in this document.

2 Verification Overview

This section provides an overview of the verification plan, process, and results of the first evaluation exercise proposed by the AURORA project in [2]: "Exercise 2-1: Verification of the methods for the on-line calculation of indicators" for the second use case "STAM process with on-line monitoring of efficiency indicators".

The objective of this verification report is to assess the feasibility of the proposed on-line flight efficiency indicator calculation methodology, and the performance of the system prototype that implements this methodology.

Specifically, the following subsections introduce the advanced performance data model with the proposed on-line indicator calculation it uses, the flight efficiency indicators that this data model computes, the testing environment and dataset used, and the results against each success criteria. Additionally, the assumptions and limitations of proposed stream-based data model are also detailed at the end of this section.

2.1 List of on-line flight efficiency indicators

This subsection provides a list of the flight efficiency indicators that are calculated using the on-line stream based data model. These 10 indicators have been developed with input from airspace users, SJU reviewers, and other AURORA project members based on [3] and discussions in subsequent workshops. In the identifier, subset, reference trajectory, and description of all 10 indicators are summarised.

Indicator	Subset	Reference Trajectory	Description
KEA	Horizontal	Geodesic	Quantifies the horizontal deviations of the actual trajectory in comparison with the geodesic trajectory.
KEA_P	Horizontal	Flight Plan	Quantifies the horizontal deviations of the actual trajectory in comparison with the planned trajectory.
KEA_C1	Horizontal	Optimal Cost-based (time and fuel)	Quantifies the horizontal deviations of the actual trajectory in comparison with the optimal cost-based (time & fuel) trajectory.
KEA_C2	Horizontal	Optimal Cost-based (time, fuel, and taxes)	Quantifies the horizontal deviations of the actual trajectory in comparison with the optimal cost-based (time & fuel & taxes) trajectory.
FEA_P	Fuel	Flight Plan	Quantifies the extra-fuel of the actual trajectory in comparison with the planned trajectory.
FEA_C1	Fuel	Optimal fuel-based	Quantifies the extra-fuel of the actual trajectory in comparison with the optimal cost-based (time & fuel) trajectory.
FEA_C2	Fuel	Optimal Cost-based (time, fuel, and taxes)	Quantifies the extra-fuel of the actual trajectory in comparison with the optimal cost-based (time & fuel & taxes) trajectory.

Indicator	Subset	Reference Trajectory	Description
CEA_P	Cost	Flight Plan	Quantifies the extra-costs of the actual trajectory in comparison with the planned trajectory.
CEA_C1	Cost	Optimal Cost-based (time and fuel)	Quantifies the extra-costs of the actual trajectory in comparison with the optimal cost-based (time & fuel) trajectory.
CEA_C2	Cost	Optimal Cost-based (time, fuel, and taxes)	Quantifies the extra-costs of the actual trajectory in comparison with the optimal cost-based (time & fuel & taxes) trajectory.

Table 2. On-line indicators calculated

2.2 On-line flight efficiency indicators calculation

This subsection provides an overview of methodology proposed to calculate flight efficiency indicators on-line for AURORA. The existing off-line calculation can obtain only one final value for each flight after its landing, while using novel on-line calculation for AURORA, all flight efficiency indicators can be accessed for all trajectory points over the duration of a flight (i.e. after flight departure and before it is landed). This on-line calculation methodology is used in the advanced performance data model tested in this study.

In general, for a certain flight, given a time stamp t_i, any one of its indicators can be computed on-line using based on a measure, M, using:

$$IND_{t_i} = \left(\frac{M_{t_i}}{M^*_{t_i}} - 1\right)\%$$

where M_{t_i} defines the actual measure (i.e. distance, fuel, overall monetary cost) at time stamp t_i, which can be calculated using surveillance data; $M^*_{t_i}$ defines its corresponding optimal measure given the same actual trajectory point at the same time stamp, which can be retrieved from pre-calculated generated (i.e. reference) trajectories using nearest point search.

For example, Figure 1 illustrates a flight traveling from Dublin to London on 20th Feb 2017. To calculate the value of the CEA_C1 indicator at the moment just before the flight's landing, t_i, the 2D location (i.e. longitude and latitude) of the actual trajectory point (from the "RECON" trajectory) at t_i is used to find the corresponding nearest point in the generated free routing reference trajectory, "FREE_CI", (as shown in the highlight in Figure 1). For these two points, the overall cost value of the actual "RECON" trajectory, as C_{t_i}, is compared to the overall cost value of "FREE_CI" reference trajectory, $C^*_{t_i}$, using the formula above.

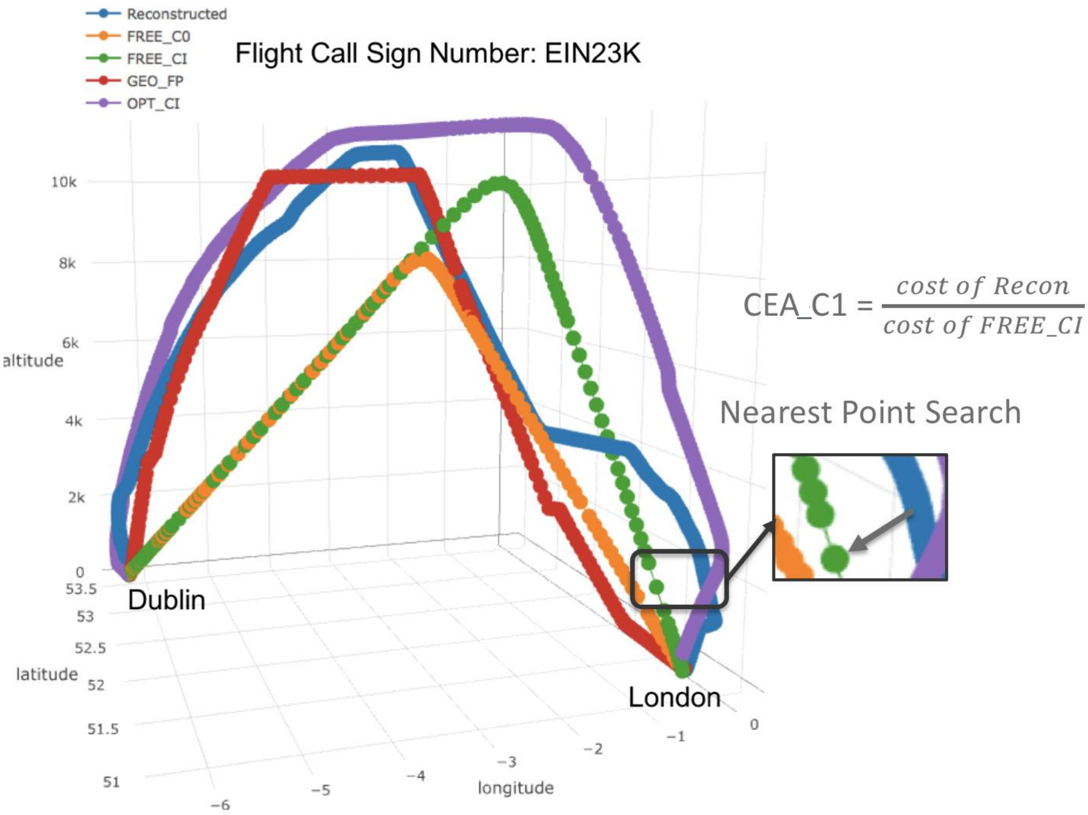

Figure 1. Example for on-line calculation of flight efficiency indicators

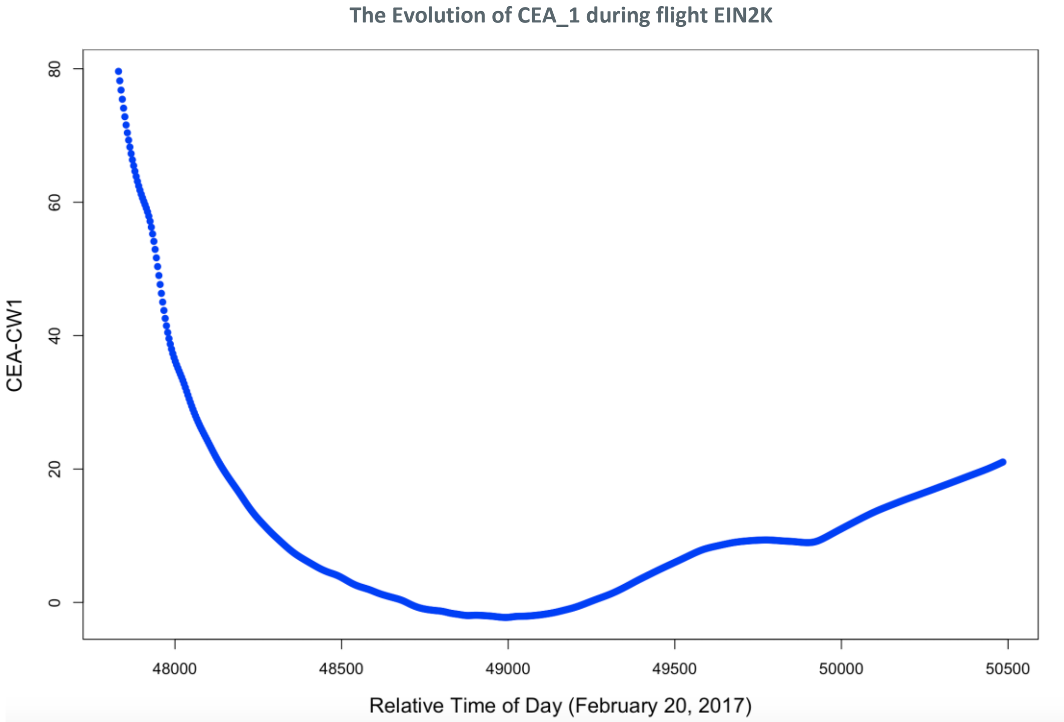

Figure 2. Example for the evolution of an on-line flight efficiency indicator

Figure 2 shows an example of the evolution the CEA_C1 efficiency indicator for the same flight over the course of the flight (i.e. call sign number: EIN23K). As shown, the efficiency at the beginning is very poor but drops sharply to a more stable level within about 10 minutes (i.e. time 47800 to 48400). Afterwards, the indicator decreases slightly its minimum at around the middle of the full flight and then increases gradually until it is landed.

2.3 Advanced performance data model

This subsection provides an overview of the advanced performance data model designed for calculating the flight efficiency indicators on-line. This overview includes the description of each data model component and the data flow. Figure 3 illustrates the architecture of advanced performance data model for AURORA.

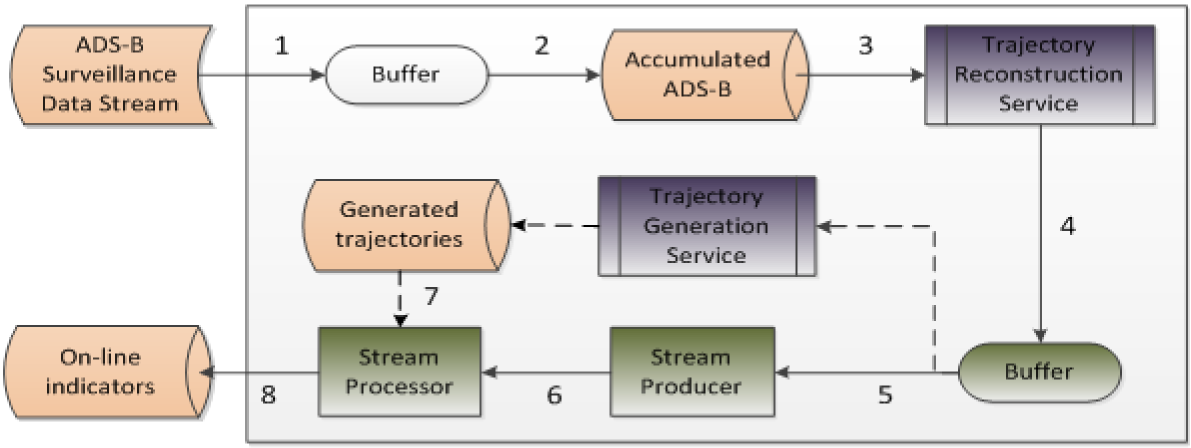

Figure 3. The architecture of advanced performance data model

The architecture illustrated Figure 3 contains the following components:

- **ADS-B surveillance data stream**: Raw ADS-B data provided by FlightRadar 24 in JSON format. This data stream updates the details of all ADS-B monitored aircrafts in the ECAC area about every 5 seconds.
- **Accumulated ADS-B**: To fit the input requirements of trajectory reconstruction service [6], the pipeline cleans, accumulates each latest ADS-B update, and organises them by each flight since its departure.
- **Trajectory Reconstruction Service**: BRTE's proprietary trajectory reconstruction service can estimate aircraft status details (including instantaneous mass) given each aircraft's ADS-B trajectory point.
- **Stream Producer**: Send reconstructed trajectory stream reliably to the stream processor, without any data loss, duplication, or out-of-sequence issues. The Producer API of Apache Kafka [4] is used for the implementation of this component.
- **Stream Processor**: Calculates flight efficiency indicators using parallel computing based on distributed system and writes the results to data store reliably, without any data loss, duplication or out-of-sequence issues. Apache Spark [5] Streaming is chosen as the technology to implement this component.

- **Trajectory Generation Service**: Reference trajectories (i.e. shortest, least-fuel, least-cost) are calculated using BRTE's trajectory generation service. All those trajectories are persisted in a database for calculating flight efficiency indicators on-line.
- **Flight Efficiency Indictors**: All flight efficiency indicators calculated by the stream based data model are persisted into a PostGIS database for more complicated data queries, for instance, checking equity indicators given a specific spatial and temporal range.
- **Buffers**: There are two buffers in the stream-based data model. The first one adjusts the flow rate between raw ADS-B data stream and the trajectory reconstruction. The second one adjusts the flow rate between the reconstruction trajectory stream, and the indicator calculation. The second buffer is implemented using Apache Kafka service, as it supports high throughput and can share the same parallelism strategy with Spark.

The data flow of this architecture is labelled with digits 1 to 8. These are explained as follows:

1. The ADS-B surveillance data stream is sent to a buffer to adapt to the receiving rate and the subsequent processing rate.
2. The contents of this buffer are then cleared and appended to the accumulated ADS-B data store which is partitioned by flight id. We use the ''call sign number" combined with ''departure time" to uniquely identify a flight.
3. The trajectory reconstruction service is triggered periodically, e.g. every 5 seconds, to derive extra states (i.e. mass) for all updated actual trajectory points. To avoid a performance bottleneck, this reconstruction service is called in a multi-threaded manner, with the unit of parallelism as each unique flight.
4. These reconstructed trajectories are sent on to an Apache Kafka buffer. This reliable buffer can ingest data with high throughput and low latency for more complicated processing tasks afterwards.
5. The Kafka stream producer reads reconstructed trajectory streams from the buffer and sends them to the stream processor and the trajectory generation service. The stream producer guarantees reliable message transmission with no duplication, no data loss, and no out-of-sequence messages. The trajectory generation service creates the reference trajectories which are stored in database.
6. The Stream Processor, which is implemented using Apache Spark Streaming, pulls the reconstructed trajectory streaming data every 30 seconds to aggregate a micro-batch and computes the efficiency indicators that correspond to all new reconstructed trajectory points, such as travelled distance, consumed fuel, and overall cost.
7. This stream processor also retrieves the relevant optimum value using **nearest point search** from pre-loaded in-memory generated trajectories data, then calculates required flight efficiency indicators with the actual value from reconstructed trajectory point. The broadcast mechanism in Spark is used for pre-loading generated trajectory data to avoid sending copies to all worker machines every time a new micro-batch is formed. The calculation so far is defined with a set of stateful transformations (rather than actions) to avoid generating large intermediate datasets.
8. The stream processor uses the ''foreach" action to finally output the calculated on-line indicator results on to PostGIS for subsequent complex queries. For example, the air traffic network manager can check the evolution of an indicator - KEA in one sector - to see if it is relatively fairly distributed among airlines.

Additionally, due to data privacy issue, the initial mass, which is required for trajectory generation, is not available for this study. This value is usually stored in quick access recorder (QAR) which is owned by airlines. Therefore, we use the estimated initial mass from the output of trajectory reconstruction, which lead to a periodically updated trajectory generation service.

2.4 Trajectory reconstruction, generation and optimization services

This section provides a brief description of the trajectory reconstruction, generation and optimization services used in steps 3 and 5 (Figure 3).

INTRAC: Aircraft Intent Inference and Trajectory Reconstruction service

The input to this process is a set of flights with associated surveillance tracks (ADS-B) and aircraft type information, for example, from flight plans. The objective of the service is to reconstruct, for each of the flights, the evolution of the aircraft state (chiefly speed and mass) from the surveillance data (ADS-B), in order to estimate the fuel consumed during the flight. To that aim, the service first builds an instance of aircraft intent that fits the flight track data using the intent inference module (Intent Inference Infrastructure in Figure 4) and then feeds the resulting aircraft intent expressed in AIDL (Aircraft Intent Description Language) to the trajectory computation module (Trajectory Computation Infrastructure in Figure 4), which integrates the full trajectory and obtains a sequence of aircraft states, including position, altitude, airspeed and instantaneous aircraft mass, from take-off to landing. The processes that occur within the service, which are schematically depicted in Figure 4, are briefly described below:

- For each flight, an instance of aircraft intent, expressed in AIDL, that fits the surveillance tracks of that flight is built. This AIDL instance includes a lateral thread, which consists of a sequence of geometric constructs (segments of geodesics and circular arcs) that match the horizontal projection of the surveillance reports (latitude/longitude coordinates), and two vertical threads, which consist of sequences of kinematic instructions (altitude and airspeed) that match the sequence of the aircraft's altitudes and airspeeds. To obtain the aircraft's airspeed from the flight track data, we use the GFS (Global Forecasting System) Atmospheric Model meteorological forecasts for the time interval in question to derive airspeeds from ground speeds. The ground speeds are obtained as derivatives of the sequence of time stamped positions.
- Aircraft mass may be estimated based on the AIDL instance obtained in the previous step and by setting the total aircraft weight at some point of the flight to a given value. If no actual weight information is available, total aircraft weight may be assumed for some point of the flight, typically at take-off or landing (e.g. landing weight equal to 120% of the Operating Empty Weight (OEW) of the aircraft type in question) and then iterating in that mass in an optimization process to obtain a final value that better fits the surveillance profile.
- The resulting AIDL instance is fed to the Trajectory Computation Infrastructure together with the initial conditions (time, mass, position, altitude and speed). To integrate the aircraft's trajectory, the Trajectory Computation Infrastructure could use BADA 3.13 or BADA 4.2 [7][8] as aircraft performance model and the GFS (Global Forecasting System) Atmospheric Model [9][10] as the model of the meteorological conditions encountered by the aircraft as it flies. In the AURORA scenarios, all the results were obtained using BADA 3.13, to maximize the aircraft coverage at the ECAC.
- The estimate of the fuel burn during the segment of trajectory considered is calculated from the trajectory output by the Trajectory Computation Infrastructure as the difference between the initial mass and the landing mass.

Depending on the quality of the data source, the surveillance tracks used as input for this process may require some post-processing to perform validation, track indexing, outlier removal, smoothing, flight plan matching, etc.

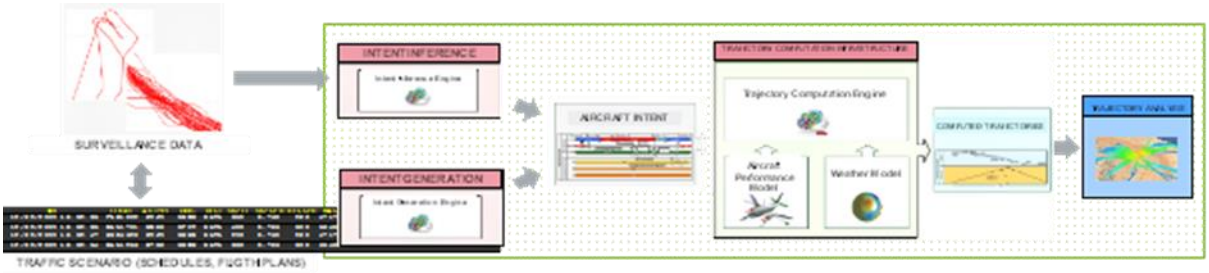

Figure 4. INTRAC: Aircraft Intent Inference and Trajectory Reconstruction Process

The initial deployment of the trajectory reconstruction service will rely on a samba-share-based file exchange mechanism where the user will place a trajectory reconstruction request file in the input directory "fr24aurora/in". The service will process the trajectory reconstruction request file and output a reply file with the reconstructed trajectory to be found in the output directory "fr24aurora/out", and a log file with the originating request file to be found in the log directory "fr24aurora/log".

Request and reply files must be in a JSON trajectory data file format. Currently, trajectory reconstruction requests can be processed only one at a time and the request trajectory state vector must contain at least 4 samples. All request and reply state vector variables are to be in the International System of Units. Request-reply pairing can be accomplished either through a "jobID" field in the JSON trajectory data file, or directly through the file name: reply and log files will have a time stamp prep-ended to the originating request file name. Should the service fail to process the trajectory reconstruction request file, it will additionally prep-end a "FAIL" tag to the reply and log files.

INCEPT: Aircraft Intent Generation and Trajectory Synthesis Service

The input to this service is a set of flights plans (obtained from DDR2) with associated initial conditions, which are the aircraft state variables (e.g., altitude, speed, position, time, etc.) at the point where the flight plan is active. The objective of the service is to generate the aircraft intent associated with these flight plans, and build the corresponding trajectory from that aircraft intent, which is in turn associated to the specific flight plan. To that aim, we first build an instance of aircraft intent that fits the all the restrictions associated to the flight plan and the operational context using the intent generation module (Intent Generation Infrastructure in Figure 5) and then feed the resulting aircraft intent expressed in AIDL to the trajectory computation module (Trajectory Computation Infrastructure in Figure 5), which integrates the full trajectory and obtains a sequence of aircraft states, including position, altitude, airspeed and instantaneous aircraft mass, from take-off to landing. The processes occurring within the service, which are schematically depicted in Figure 5, are briefly described below:

- For each flight plan associated with each flight, build an instance of aircraft intent expressed in AIDL (Aircraft Intent Description Language) that complies with the route and restrictions included in the flight plan. In addition, this AIDL will have to comply with all the constraints and procedures included in the operational context in which the flight plan is active. The initial conditions can be extracted from the surveillance analysis done by the trajectory reconstruction service up to a specific point in a trajectory or provided manually as an input.

- The resulting AIDL instance is fed to the Trajectory Computation Infrastructure together with the initial conditions (time, mass, position, altitude and speed). To integrate the aircraft's trajectory, the Trajectory Computation Infrastructure uses BADA 3.14 or BADA 4.2 as aircraft performance model and the GFS forecasts as the model of the meteorological conditions encountered by the aircraft as it flies. In the AURORA scenarios, all the results were obtained using BADA 3.10, to maximize the aircraft coverage in the ECAC.
- The estimate of the fuel burn during the segment of trajectory considered is calculated from the trajectory output by the Trajectory Computation Infrastructure as the difference between the initial mass and the landing mass.

Alternative scenarios can be tested by changing the different input data that feed the service:

- Operational context data to test different airspace set ups.
- Weather conditions to study its impact on the input traffic data.
- Initial conditions to explore different starting conditions for the traffic

Figure 5. INCEPT: Aircraft Intent Generation and Trajectory Synthesis Process

The generation process runs inside BOEING's servers in a machine with 16 cores and 47GB of RAM. It relies on a samba file exchange system where the user should load the necessary input files (flight intent, initial conditions, weather model and operational context if wanted). To start the service the user needs only to place a 'batch' file in the batch folder inside the samba structure of folders. This 'batch' is an xml file containing all the routes to the different inputs, outputs and also the aircraft BADA model to be used. The service will run the trajectories in parallel using all cores available, once the service is done with one batch, this will be moved to the completed folder, and the service will start the computation of the next batch in the folder. Typical calculation times are around 30 seconds per trajectory, in case a trajectory fails to generate, in order to avoid long calculation times, there is a 3 minute time-out.

INTRO: Intent-based Trajectory Optimization Service

The objective of the optimization service is to obtain different sets of optimal trajectories (optimal fuel trajectories or optimal cost trajectories) which are the result of minimizing a cost function (Cost = fuel consumption + CI x flight time). To achieve this, the flight intent of the required trajectory must be given together with an initial guess, a trajectory that fulfils all the boundary conditions of the problems (with the same aircraft, weather and earth model) but is not necessarily optimal. In case of not having an initial guess, the service could calculate one. The optimization is a two-step process schematically shown in Figure 6:

- First we write and solve the optimization problem. This is done with the OptGen library, which writes a nonlinear optimization problem and uses the IPOPT algorithm to solve it.

- Then we write the solution in the form of AIDL instances using the IntSynth library. These instances will then be fed to the Trajectory Predictor to finally obtain the optimal trajectory.

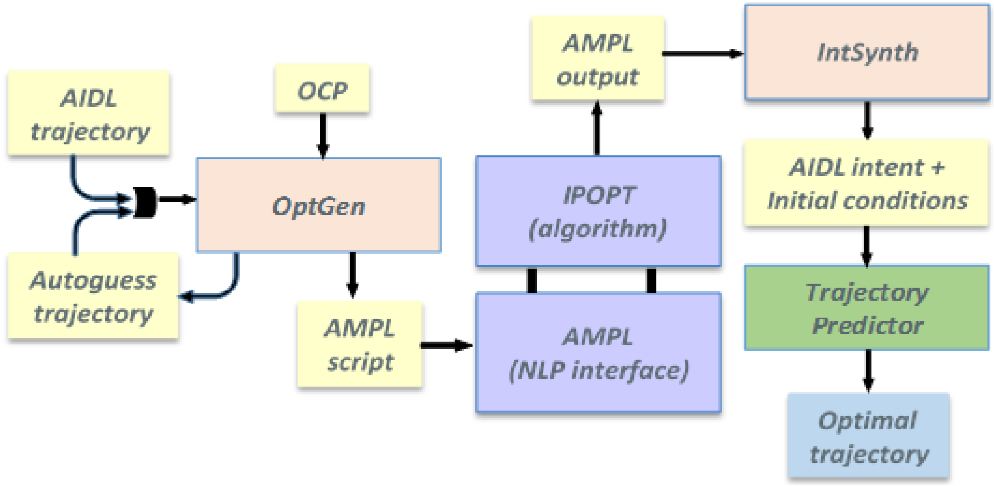

Figure 6. Architecture of the optimization infrastructure

The trajectories obtained may vary depending on the user's input:

- If a Cost Index is given, the solution will be an optimal cost trajectory. If no Cost Index is given, the solution will be an optimal fuel trajectory (as expressed in the Cost function).
- The process can also give two types of trajectories depending on the lateral path. If we want the algorithm to optimize the route of the trajectory from one point to another (considering the weather conditions), we can select the free routing option. If we want the trajectory to follow a specific route, we can select the optimal path option.

This service runs in a machine in BOEING's servers with 8GB of RAM memory and 8 cores. To run the service we must only log in to the server, upload the flight intent files (and the initial guess trajectories if desired), and start the service via a command where the path to the flight intents and the type of trajectories to calculate (free routing or optimal path, with or without CI) must be given. The process will calculate the trajectories one by one. Typical calculation times for the optimal trajectories are 2 minutes.

2.5 Experimental assumptions and limitations

Having described the stream-based data model, this subsection outlines some limitations in the current implementation used for testing, as well as describing assumptions made to draw conclusions from the testing results.

The main limitation is that the trajectory reconstruction and generation services were not implemented in a way to allow their integration into the stream data model implementation. Currently, to reconstruct or generate a trajectory, requests need to be generated in JSON or XML format files, and posted to the server on which the BRTE trajectory services run. The BRTE trajectory services will then be automatically triggered and the output reconstructed or generated trajectories will subsequently become available in JSON or XML format files on the server. While the processes of generating and reconstructing trajectories run within the limits required by the online data streaming platform the process of calling them requires too many time-consuming disk input / output operations to be fast enough to integrate with the streaming data pipeline.

This performance limitation can be relatively easily solved in two ways. First, the method for calling the BRTE trajectory services could be changed from one based on exchanging files on disk to a remote process call implemented through a Restful API. This would remove the need for time-consuming disk I/O operations. Second, more parallel instances of the reconstruction and generation services could be added on multiple servers so that the processes could be parallelised (there is no reason not to parallelise these processes). However, the timeline of the AURORA project did not allow for these solutions to be put in place before testing.

To evaluate the stream-based data model without actually integrating the BRTE trajectory services we simulated calls to the reconstruction services with reasonable time delays for these processes and used precomputed trajectories as output. The input of this simulator is surveillance trajectory points, the output is instantaneous mass for each given surveillance trajectory point by reading reconstructed trajectory which is pre-processed off-line for this simulation. The delay that this reconstruction process contributes to the whole stream-based data model is dependent on the number of requests (i.e. flights) with the following assumptions:

1. The simulated delay for running the drawn randomly from a normal distribution with mean 1.0 and standard deviation 0.1 (this matches the time that the actual reconstruction process takes)
2. The total amount of time spent = time spent on each request * CEIL(the number of flights / the number of parallel units in server);
3. The number of parallel units is set as the number of CPU processors each server (24) * the number of cores each CPU processor (6) = 144. This means the amount of time spent by reconstruction service for 1 flight would be the same for reconstructing 144 flights, would be a half for reconstructing 145 flights.

While the trajectory generation service was capable of generating optimal trajectories on average in 2 minutes the requirement of excessive disk input/output operations which added to this time again made it impossible to integrate current implementation with the streaming platform due to the number of trajectories that needed to be generated - 5 optimal trajectories per flight for each of the flights per test day (between 13,000 and 15,000). This limitation could easily be remedied if multiple parallel instances of the trajectory generation service were made available but unfortunately this was not possible within the scope of the AURORA project and so a simulation approach was used. All optimal trajectories were pre-computed and made available within the stream-data pipeline (e.g. every 3 hours).

Another limitation was that the full data cleansing process was not integrated into the stream-based model as the process should fit all potential requirements of trajectory reconstruction service input which is not clearly defined so far, except for "no duplicated data" and "no out-of-sequence data".

Finally we should mention that it was only possible to generate FREE_CI and OPT_CI trajectories for flights between 12:00-14:00 on both 20th and 24th Feb traffic samples. This is identified as a limitation of the generation processes because it is related to performance issues of BRTE trajectory optimisation service that could be easily solved by increasing the computation power. More details on this can be found in [3]. In any case, this is not considered a significant limitation of the on-line data model as it has no major effect on the on-line calculation process.

2.6 Testing environment and dataset

This subsection provides the specifications of the testing environment and the testing datasets.

The on-line stream based data model implementation is tested on a local machine with the following specifications:

- CPU: Intel Core i7K 8700K (6-Core/12-Thread) 3.7 GHz
- Memory: 32GB Dual Channel DDR4 at 2666MHz
- Storage: 2TB 7200RPM SATA 6Gb/s

The versions of key software on this machine are:

- Operating system: Ubuntu, version 16.04
- Spark: Spark 2.2.1 (scala: 2.11)
- Kafka: 0.11.0.2 (scala: 2.11)
- PostgresSQL: 9.6.5

The testing datasets used in this study use two traffic samples, all ADS-B monitored flights departs and lands in ECAC area on 20th Feb, 2017 and 24th Feb, 2017, respectively. The numbers of generated trajectories in RECON, GEO_FP and UP are not comparable with FREE_CI and OPT_CI as the traffic samples are different as it is explained in previous section.

	RECON	GEO_FP	UP	FREE_CI	OPT_CI
20th Feb, 2017	13,836	13,978	13,879	1,326	1,155
24th Feb, 2017	15,189	15,423	15,381	1,512	1,319

Table 3. The number of trajectories (flights) of both testing traffic scenarios

As well as using trajectory data our experimental scenarios share the same dataset for other aspects required using computing indicators - for example, maximum take-off weight, cost index, the partition and rate for each airspace in ECAC area, and oil price (0.4638 €/kg).

The data format of reconstructed trajectories (RECON) used in the platform is:

> flight id (departure time + call sign number), timestamp, longitude, latitude, altitude, instantaneous mass

The data format of all generated reference trajectories (GEO_FP, UP, FREE_CI, OPT_CI) used in the platform is:

> flight id (departure time + call sign number), timestamp, longitude, latitude, altitude, cost in distance(travelled distance), cost in fuel (consumed fuel), cost in euros (i.e. monetary cost includes time, fuel, and route charge)
>
> The number of FREE_CI and OPT_CI trajectories are less than other types of trajectories. This is because BRTE only generates FREE_CI and OPT_CI trajectories for flights between 12:00-14:00 on both 20th and 24th Feb traffic samples, which is caused by the performance issues of their existing trajectory optimisation service. More details on this can be referred to [3]. It is not considered as a main limitation of on-line data model as it has no major effect on the on-line calculation process. The potential increase of FREE_CI and OPT_CI to the equivalent amount of other trajectories types should only lead to slower update of generated trajectories, which can be easily solved by slightly increase the amount required memory space (i.e. by approximately 4GB in total), rather than the change of the whole system architecture.

2.7 Summary of verification results

This subsection summarises the verification results in Table 4, Table 5, Table **6** and Table 7.

Verification Objective Id	**VER-OBJ-ONLINE-2.1.1**
Verification Objective Description	Read and process data in a streaming context.
Success Criteria 2.1.1-1	The data streamed into the application does not exceed the minimum threshold of the message loss rate (Ideally, it would be zero).
Exercise Results	For the cleaned testing traffic sample on Feb 20, 2017, the number of input data surveillance points is 17,249,171 and the number of output indicator points is 17,249,171.
Objective Status	Succeed
Success Criteria 2.1.1-2	The data is cleaned enough to make sure there are no duplicate data points that can derive in errors or deviations in the on-line calculation.
Exercise Results	For the testing traffic sample on Feb 20, 2017 and Feb 24, 2017, the raw ADS-B data has 24.91% and 26.29% duplicated data records, respectively. For the cleaned dataset on both dates, 0 duplicated data records are present.
Objective Status	Succeed
Success Criteria 2.1.1-3	The data is sorted by timestamp.
Exercise Results	For the testing traffic sample on Feb 20, 2017 and Feb 24, 2017, the raw ADS-B data has 40.58% and 40.82% out-of-sequence data records, respectively. The cleaned datasets on both dates have 0 out-of-sequence data records.
Objective Status	Succeed

Table 4. Verification results for objective VER-OBJ-ONLINE-2.1.1

Verification Objective Id	**VER-OBJ-ONLINE-2.1.2**
Verification Objective Description	Calculate the flight efficiency indicators accurately along the flight trajectory.
Success Criteria 2.1.2-1	The indicators calculated on-line match those calculated off-line for the whole trajectory (from origin to destination).
Exercise Results	For the whole flight trajectory, the maximum absolute error among all 10 flight efficiency indicators when comparing their values obtained by the on-line calculation method and existing off-line method, is less than 5 percentage points for both traffic samples.
Objective Status	Succeed
Success Criteria 2.1.2-2	The method to calculate off-line (trajectories generation, reconstruction and optimal routes) is valid to obtain the efficiency indicators for the actual position of the aircraft.
Exercise Results	For all input actual positions of aircraft, has all 10 flight efficiency indicators calculated.

Objective Status	Succeed
Success Criteria 2.1.2-3	The method to calculate off-line (trajectories generation, reconstruction and optimal routes) is valid to obtain the predictions of the efficiency indicators at destination when the aircraft is at a certain point of the trajectory.
Exercise Results	This criterion was removed from the scope of the project.
Objective Status	N/A.

Table 5. Verification results for objective VER-OBJ-ONLINE-2.1.2

Verification Objective Id	**VER-OBJ-ONLINE-2.1.3**
Verification Objective Description	Calculate indicators efficiently enough to match specifications of AURORA STAM use cases i.e. the method to calculate off-line (trajectories generation, reconstruction and optimal routes) in combination with advanced analytics is feasible to obtain the efficiency indicators online.
Success Criteria 2.1.3-1	The level of latency in processing the streaming data matches the requirements of the AURORA STAM use case.
Exercise Results	As suggested from AUs workshop, a latency rate for calculating indicators is 5 minutes. The current prototype can respond with flight efficiency indicators within one minute.
Objective Status	Succeed
Success Criteria 2.1.3-2	The throughput achieved in processing the streaming data matches the requirements of the AURORA STAM use case.
Exercise Results	The throughput rate achieved by the system is inline with the throughput rates expected by AU users.
Objective Status	Succeed

Table 6. Verification results for objective VER-OBJ-ONLINE-2.1.3

Verification Objective Id	**VER-OBJ-ONLINE-2.1.4**
Verification Objective Description	Select the most appropriate streaming data technology for the AURORA STAM use case.
Success Criteria 2.1.4-1	The technology selected has achieved the highest score based on a linear combination of latency, throughput, message loss rate and system cost (computing and memory). The importance of these factors is determined by the AURORA STAM use case.
Exercise Results	Apache Kafka and Spark Streaming are selected.

Verification Objective Id	**VER-OBJ-ONLINE-2.1.4**
Objective Status	Succeed.
Success Criteria 2.1.4-2	STAM use case requirements are covered by the selected data streaming technique.
Exercise Results	All STAM use cases within the scope of the project are covered by the selected data streaming technique.
Objective Status	Succeed.

Table 7. Verification results for objective VER-OBJ-ONLINE-2.1.4

2.8 Deviations from the planned activities

There are four main deviations from the planned activities:

- Originally it was intended to also use a traffic sample from 10th Jan, 2017, however, this sample only contained approximately one thousand flights, which is not enough to conduct performance testing of stream based data model. Instead, we use 20th Feb 2017, and 24th Feb 2017 air traffic data to keep consistent with our previously related deliverables.
- The proposed prediction functionality was deemed out of scope of the project.
- The trajectory reconstruction and generation service are not actually called but instead are simulated. These services are proprietary implementations owned by BRTE and the current implementations cannot match the throughput and latency requirements of the online platform.
- For the testing methodology, the implementation of stream based data model is not compared with other frameworks as from the literature the existing framework Apache Kafka + Apache Spark is the de-facto standard in big data streaming technology and met the required latency and throughput criteria for the AURORA project.

3 Verification Results

This section outlines the detailed verification results, against all success criteria for each of the four verification objectives proposed in [2].

3.1 Verification results of VER-OBJ-ONLINE-2.1.1

This verification objective addresses "*Read and process data in a streaming context*" through three success criteria identified in [2]. To verify if success criteria 2.1.1-2 and 2.1.1-3 are met, two datasets are assessed for each traffic scenario, raw ADS-B data from FR24, and cleaned ADS-B data from BRTE that causes no error when invoking the trajectory reconstruction service.

Success Criteria 2.1.1-1: "*The data streamed into the application does not exceed the minimum threshold of the message loss rate (Ideally, it would be zero)*".

Message loss rate is defined as the number of data points lost over the stream-based data model, in particular, from when the surveillance points are being received by trajectory reconstruction service, to the moment when the these surveillance points with their calculated flight efficiency indicators values are persisted in the database (i.e. from step 4 to step 8 as shown in Figure 3). Because the previous steps are related to data cleansing, this loss rate would indicate how reliable the connection of the data stream is. The higher the value of this loss, the less reliable is the connection. Note that message loss rate does not correspond to the reduction of data after the data cleaning process (i.e. from step 1 to step 3 in Figure 3). Data cleaning is required so that all input data would meet the requirements of trajectory reconstruction service, for example, no duplicated and out-of-sequence data in the pipeline.

Since the on-line system uses Apache Kafka and Spark Streaming, which are designed to be fault tolerant and robust, these frameworks ensure that the message loss rate (over the network) is kept to a minimum or zero. Based on the tests of the on-line system performed on the data for two days, 20th February 2017 and 24th February 2017, it was found that there were no data points lost over the proposed stream-based data model. All the points that were fed into the system were processed, either by means of data cleaning or processing it on-line. This indicates that the connection to the stream was consistently strong.

Hence, for all the data points on both days, 20th and 24th February 2017, **message loss rate = 0.**

Success Criteria 2.1.1-2: "*The data is cleaned enough to make sure there are no duplicate data points that can derive in errors or deviations in the on-line calculation*".

The duplicated data record is identified when the aircraft identifier, location, and timestamp combined in multiple surveillance points are found the same. Before the on-line calculation of efficiency indicators, the raw ADS-B data needs to be ingested into the data pipeline. This data cannot be directly used in the system as it may contain duplicates. Hence this raw data is checked using a script that detects the number of duplicated data points on the raw ADS-B data on 20th and 24th February 2017.

In the script, first, the raw ADS-B JSON files are parsed and filtered so that only the data fields required to check whether there are duplicates remain. These fields are: *Aircraft_ID, Latitude, Longitude, Altitude,* and *Updated_Timestamp*. A combination of these values is sufficient to check the duplicates. This is because for each aircraft ID, it is not possible that the combination of its location, altitude and timestamp are exactly the same. After filtering, a set is created containing all the unique combinations of the values of the data fields. The number of records in this set is then subtracted from the total number of points which results in the number of duplicates.

This script was run on the raw and cleaned ADS-B data on two days: 20th February 2017 and 24th February 2017. These results are summarized in Table 8.

Dataset	Total Data Points	Number of Duplicates	Percentage
20th Feb 2017 - Raw	49,654,752	12,368,589	24.91%
20th Feb 2017 - Cleaned	17,249,171	0	0.0%
24th Feb 2017 - Raw	53,721,217	14,179,815	26.39%
24th Feb 2017 - Cleaned	18,677,424	0	0.0%

Table 8. Comparison of data duplication for raw and cleaned ADS-B datasets

On the 2 days, approximately 25% of the data is duplicated and is hence cleaned before it is ingested into the on-line streaming pipeline. After all data is cleaned by BRTE, they all have no duplicates.

Success Criteria 2.1.1-3: "*The data is sorted by timestamp*".

Out-of-sequence data is measured in this study when the timestamp of two consecutive surveillance points are sorted in descending order. This means that a surveillance point that should arrive earlier, is actually received later. Data that is streamed into the data pipeline should be in the correct order and updated sequentially. The raw ADS-B data that was provided did not have completely sequential data and it had to be first sorted according to the timestamp.

To check whether a data point is out-of-order, its timestamp needs to be equal to or greater than the previous data point ingested. For this, a script was written to check the number of such occurrences for the raw and cleaned ADS-B data on February 20th and February 24th, 2017. The script reads all the ADS-B JSON files and within each file, the *updated_timestamp* field is noted and subtracted from the next data point's timestamp. If the result is zero or positive, then it is in order otherwise if it is negative, then it is out of order. The results obtained from the script are summarized in Table 8.

Dataset	Total Data Points	Out of Order Points	Percentage
20th Feb 2017 - Raw	49,654,752	20,151,800	40.58%
20th Feb 2017 - Cleaned	17,249,171	0	0%
24th Feb 2017 - Raw	53,721,217	21,932,092	40.82%
24th Feb 2017 - Cleaned	18,677,424	0	0%

Table 9. Comparison of data out-of-sequence for raw and cleaned ADS-B datasets.

From the results it can be seen that a large amount of raw ADS-B data (approximately 40%) is out-of-sequence and hence needed to be sorted in order to be processed by the on-line streaming system.

However, as only the "order" not the "deviation" is measured, in practice the raw data may not be delivered with excessive delay in the most cases.

The cleaned data has no out-of-sequence cases detected thus this success criteria is meet.

3.2 Verification Results of VER-OBJ-ONLINE-2.1.2

This verification objective addresses "*Calculate the flight efficiency indicators accurately along the flight trajectory*" through three success criteria identified in [2].

Success Criteria 2.1.2-1: "*The indicators calculated on-line match those calculated off-line for the whole trajectory (from origin to destination)*".

Flight efficiency indicators calculated on-line and off-line may deviate slightly due to some approximations made to accelerate on-line computation. For example, to calculate geographical distance between two locations, a quicker great circle distance function, Haversine's formula, is used rather than the more accurate but much slower Vincenty's Formula [11]. Moreover, the existing off-line calculation method varies slightly with different flight stages (i.e. take off, cruise, landing), while the on-line calculation method does not differentiate on this basis. Last but not least, the off-line results are based on verified flight trajectory data, while the on-line results use real-time data directly which inevitably contains some errors. This success criteria is to verify if deviations incurred is acceptable.

There is only one value per flight for off-line flight efficiency indicator results, as they are calculated for the whole flights trajectory after it is landed. While in the on-line platform there are an equivalent number of flight efficiency indicator values to the total number of trajectory points of a particular flight, thus only the last value of on-line flight efficiency indicators is chosen to compare.

The absolute error is used to measure these deviations, as it is the most intuitive metric to show deviations. The less its value is, the better is performance proposed data model.

For the ECAC traffic sample on 20th Feb 2017, there are 432 flights that have their off-line flight efficiency indicator values consolidated with airspace users. For the ECAC traffic sample on 24th Feb 2017, there are 783 such flights. For each flight and each efficiency indicator, its corresponding absolute error is calculated and shown in box plot separated by each indicator in Figure 7 and Figure 8. The maximum absolute error among all indicators and flights is below 3.5 percentage points, which discussions among AURORA members and airspace users at AU workshops have deemed acceptable deviation. The results of 24th Feb traffic sample have a slightly better accuracy.

The reason why the absolute error is different among indicators is the various reference trajectories used for indicator calculation. The general trend shared between the two traffic scenario samples are: the indicators that use the "c2" (i.e. optimal trajectory includes fuel, distance, and tax) reference trajectory have the highest absolute error; the indicators that use the "p" (i.e. planned trajectory) reference trajectory have the lowest absolute error. This means to detect any cost deviations from flight plan, on-line calculation is very close to existing off-line calculation. One possible reason to account for is that the last point of various reference trajectories per flight might not be perfectly matched with actual reconstructed trajectory.

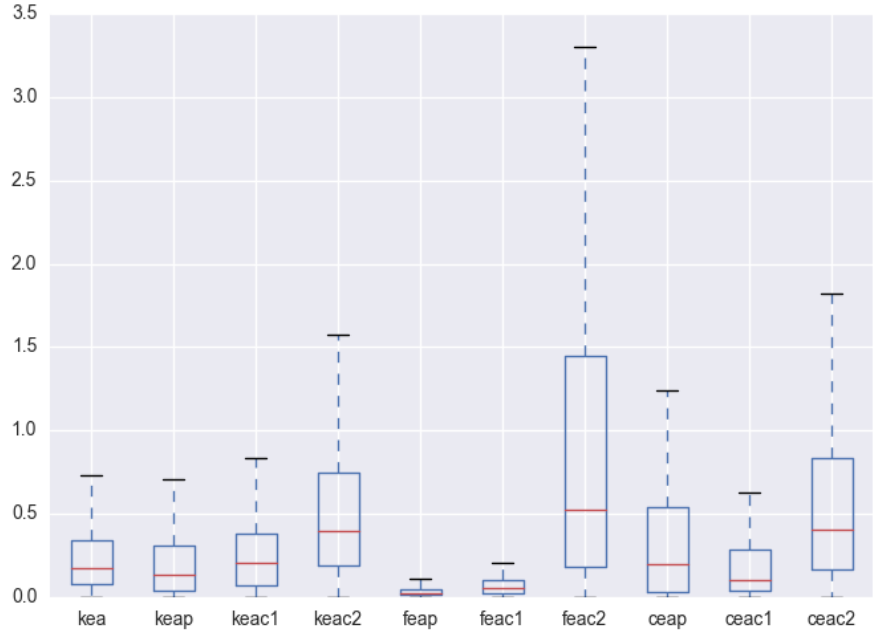

Figure 7. A boxplot of absolute error of 10 consolidated on-line flight efficiency indicators (based on ECAC traffic, 20th Feb 2017).

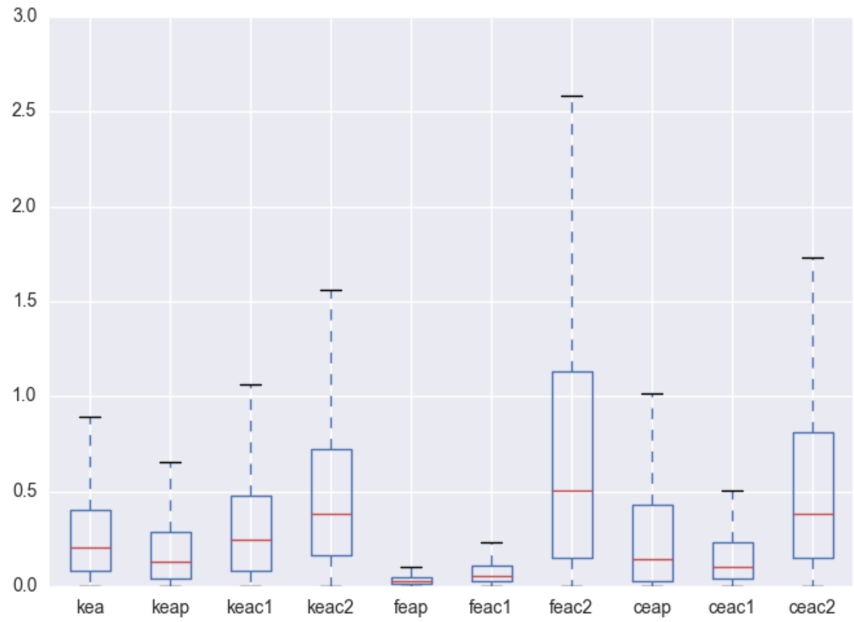

Figure 8. A boxplot of absolute error of 10 consolidated on-line flight efficiency indicators (based ECAC traffic, 24th Feb 2017).

Success Criteria 2.1.2-2: "*The method to calculate off-line (trajectories generation, reconstruction and optimal routes) is valid to obtain the efficiency indicators for the actual position of the aircraft*".

Thanks to state-of-the-art big data streaming technology (i.e. Apache Kafka and Apache Spark) used for implementing proposed data model, a zero message loss rate can be achieved to meet this success criteria, which is given each update aircraft trajectory point, stream data model can offer its corresponding flight efficiency indicators' values. For example, for 18,773,952 reconstructed trajectory points input, there are exactly 18,773,952 corresponding data points in the sink database ready for further query.

If the required reference trajectory is available, then the calculation uses the corresponding cost value from the nearest point of the given reconstructed trajectory point. Conversely, a default indicator value is set as 0.0. It will not confuse the data user, as it is impossible for any aircraft at any time point (except for the very first one) to fly exactly along its optimal route.

Last but not least, as the initial mass is not accessible from airline companies, under assumptions above, the simulator of trajectory reconstruction service is used to estimate the initial mass periodically, and then updates the generated optimal reference trajectory use the on.

Success Criteria 2.1.2-3: " *The method to calculate off-line (trajectories generation, reconstruction and optimal routes) is valid to obtain the predictions of the efficiency indicators at destination when the aircraft is at a certain point of the trajectory*".

This part of the evaluation was deemed outside of the scope of the AURORA project as there is no prediction algorithm developed in AURORA.

3.3 Verification Results of VER-OBJ-ONLINE-2.1.3

This verification objective addresses "*Calculate indicators efficiently enough to match specifications of AURORA STAM use cases i.e. the method to calculate off-line (trajectories generation, reconstruction and optimal routes) in combination with advanced analytics is feasible to obtain the efficiency indicators online*" through two success criteria identified in [2].

Success Criteria 2.1.3-1: "*The level of latency in processing the streaming data matches the requirements of the AURORA STAM use case*".

The latency is defined as the amount of time (i.e. usually in seconds) taken for a data record (in this point a trajectory point) to travel from being received by trajectory reconstruction service, to the moment when the record with associated flight efficiency indicator values are persisted in the database (i.e. from step 4 to step 8 as shown in Figure 3).

Note that this latency value also includes the simulated delay when calling trajectory reconstruction service. The simulated delay follows a normal distribution with a mean value of 1.0 second and standard deviation of 0.1 seconds. These parameters were determined in consultation with BRTE.

Figure 9 and Figure 10 show the variation of latencies, aggregated over 30 minute intervals, for both traffic scenario days. As shown in Figure 9 and Figure 10, for most time over a full day of testing traffic, the latency is more or less stable at around 25 seconds. There are a few exceptions, for example, at 15:00 on 20th Feb, and at 21:00 on 24th Feb. The reason for these abnormal variations is that every 3 hours an update is made to reference trajectories (i.e. as can be seen from both figures the about every 3 hours the error bar becomes slightly bigger). During this update, to keep data consistency, the whole pipeline stop calculating indicator values until all required data completes its pre-loading process.

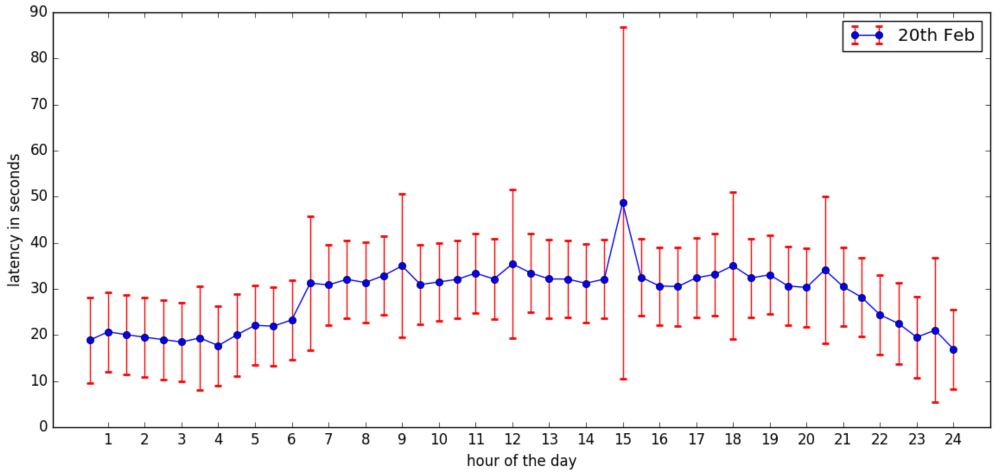

Figure 9. The error bar (standard deviation) plot of latency every 30 minutes over the full day traffic scenario on 20th Feb

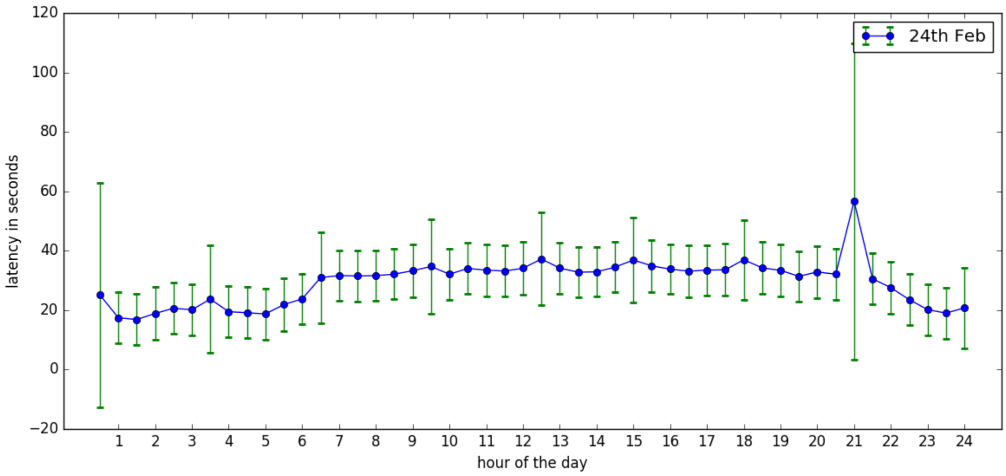

Figure 10. The error bar (standard deviation) plot of latency every 30 minutes over the full day traffic scenario on 24th Feb

The results of a further study on latency distribution are summarized in Table 10 with some key statistical metrics. Although the maximum delay for a single message on both traffic samples can be up to 6 minutes, 99% of messages can still be successfully delivered with calculated flight efficiency indicators less than 80 seconds. The mean and median latency is about only 30 seconds. Considering the target set during AURORA airspace users workshop of 5 minutes, this success criteria is meet.

	mean	median	min	max	5%	95%	99%
20th Feb	31.85	31.05	1.21	200.98	16.39	46.36	73.37
24th Feb	33.20	32.14	1.19	368.67	17.34	47.83	77.36

Table 10. Main statistics of latency results on both traffic scenarios

Success Criteria 2.1.3-2: "*The throughput achieved in processing the streaming data matches the requirements of the AURORA STAM use case*".

Throughput is defined as the number of trajectory points that come into the data pipeline in a period of time. This study measures the data pipeline throughput as the data rate after a surveillance data point is reconstructed.

The throughput of reconstructed trajectory stream achieved in both testing traffic scenarios is shown in Figure 11. As can be seen from this figure, both testing scenarios have a similar stream pattern over a day: starting a very low traffic volume at the beginning of a day, it increases sharply from 05:00 to 07:00, then maintains a stable peak traffic for about 14 hours, before dropping back for the last 3 hours of the day. The traffic scenario of 24th Feb is slightly bigger than the 20th Feb. This can also be found from the main statistics of throughput on both samples shown in Table 11.

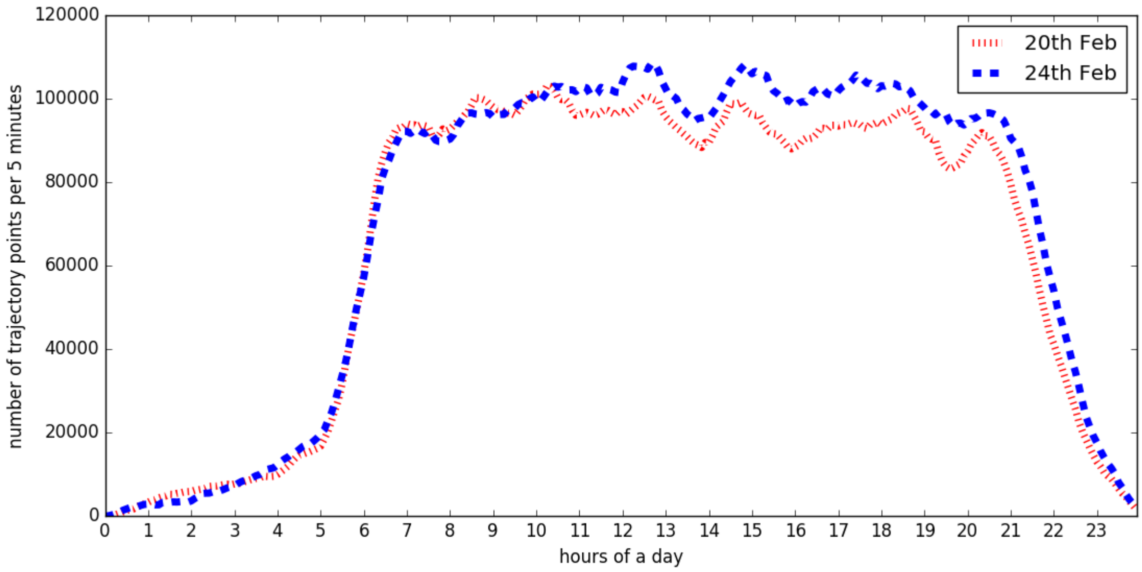

Figure 11. Throughput of stream data model for both testing traffic scenarios

	mean	median	max	25%	75%
20th Feb, 2017	65,187	90,130	103,150	16,069	95,777
24th Feb, 2017	69,363	94,764	108,391	19,036	101,185

Table 11. Main statistics of throughput achieved.

In a word, a conclusion can be drawn that the throughput achieved in processing the streaming data successfully matches the requirements of the AURORA STAM use case. In particular, the highest throughput the stream data model can cope with is 108,391 messages per 5 minutes, this is roughly equivalent to 361 messages per second. This is more than sufficient to cope with ECAC traffic levels.

3.4 Verification Results of VER-OBJ-ONLINE-2.1.4

This verification objective addresses " *Select the most appropriate streaming data technology for the AURORA STAM use case*" through two success criteria identified in [2].

Success Criteria 2.1.4-1: "*The technology selected has achieved the highest score based on a linear combination of latency, throughput, message loss rate and system cost (computing and memory). The importance of these factors is determined by the AURORA STAM use case*".

The chosen frameworks in the current implementation, Apache Kafka and Apache Spark, are now considered as the de-facto standards for big data streaming technology and met the latency and throughout requirements for the project. Therefore no other options were evaluated.

ID	Name	Specification	Source of data
REQ-STAM-001	Inclusion of initial mass	The methods for the reconstruction and generation of actual and optimal trajectories are able to consider the initial mass of the flight as an input.	Airspace users' internal databases
This requirement is **covered** by the selected data streaming technique as the initial mass is estimated using the trajectory reconstruction service.			
REQ-STAM-002	Consolidated efficiency target at destination	Systems shall be able to calculate an efficiency target per flight based on the last filed flight plans covering the whole trajectory (from origin to destination).	DDR2 (European Airspace)
This requirement is **covered** by the selected data streaming technique by integrating the trajectory generation service.			
REQ-STAM-003	Status of efficiency values	Systems shall be able to calculate the current efficiency of a flight and compare with the target at a certain point of the trajectory. Time step between subsequent calculations shall be determined by the potential deviations of the efficiency indicators with time.	FlightRadar24 (ADS-B) up to actual aircraft position
This requirement is **covered** by the selected data streaming technique as all 10 consolidated flight efficiency indicators' values can be calculated for any given ADS-B surveillance point.			
REQ-STAM-004	Predicted efficiency values at destination	Systems shall be able to calculate the expected efficiency at destination taking into consideration the flown segment of the trajectory and the foreseen trajectory. Time step between subsequent calculations shall be determined by the potential deviations of the efficiency indicators with time.	FlightRadar24 (ADS-B) up to actual aircraft position DDR2 (European Airspace)
This requirement is **not covered** as it was deemed out of the scope of the project.			
REQ-STAM-005	Efficiency alerts	Systems shall be able to raise an alert when the value of an indicator is out of the acceptable tolerance. The alert shall include which indicator value is out of the tolerance.	Tolerance levels
This requirement is **covered** by the selected data streaming technique. Although this alert system is not implemented this functionality is supported.			
REQ-STAM-006	Deviations of Efficiency targets	The system shall quantify the deviation of each indicator from the agreed value when an alert is raised.	Tolerance levels
This requirement is **covered** by the selected data streaming technique. Although this alert system is not implemented this functionality is supported.			
REQ-STAM-007	Efficiency indicators for STAM measures	The system shall be able to calculate those efficiency indicators that may change when implementing a specific STAM measure.	FlightRadar24 (ADS-B) up to actual aircraft position CFMU hotspots Pre-defined STAM measures

ID	Name	Specification	Source of data
			Flight Plans after implementing STAM measures through DDR2 (European Airspace)
This requirement is **covered** by the selected data streaming technique as all 10 consolidated flight efficiency indicators' values can be calculated at any given ADS-B surveillance point.			
REQ-STAM-008	Status of Equity & Fairness values	The system shall provide a wider view of how fairly the inefficiencies in the system are distributed among the different airspace users along the day of operation in a certain area.	FlightRadar24 (ADS-B) up to actual aircraft position
This requirement is **covered** by the selected data streaming technique in the way that all real time flight efficiency indicators' values are persisted in a database that supports spatial index, so that it can quickly response the expected equity and fairness value, once the corresponding query range (i.e. region, time range, group of flights, city pairs, etc.) is formed.			
REQ-STAM-009	Most appropriate STAM measure	The system shall be able to identify the most suitable STAM measure by balancing current ATFCM metrics (i.e. occupancy, demand) with Efficiency and Equity & Fairness indicators.	CFMU hotspots Pre-defined STAM measures Values of current and new efficiency and equity indicators
This requirement is **covered** by the selected data streaming technique. Although this alert system is not implemented this functionality is supported.			
REQ-STAM-010	Status after STAM implementation	The system shall be able to implement the agreed STAM and allow the actor to monitor the changes and evolution of indicators.	FlightRadar24 (ADS-B) up to actual aircraft position Flight Plans after implementing STAM measures through DDR2 (European Airspace)
This requirement is **covered** by the selected data streaming technique all 10 consolidated flight efficiency indicators' values can be calculated at any given ADS-B surveillance point, regardless of STAM implementation.			

Table 12. An assessment of the current online platform's alignment with the STAM use case

<u>**Success Criteria 2.1.4-2**</u>: "*STAM use case requirements are covered by the selected data streaming technique*".

The assessment of the current online platform's alignment with the STAM use case is detailed in Table 12**¡Error! No se encuentra el origen de la referencia.** Overall, 9 of 10 STAM use case requirements are covered by the selected data streaming technology. The STAM use case not covered was deemed out of scope of the project.

4 Conclusions and Recommendations

4.1 Conclusions

Overall, the testing of the advanced performance data model has been successful. It has been shown that it is technically feasible to calculate performance indicators on-line. The experimental plan specified four verification objectives:

- **VER-OBJ-ONLINE-2.1.1 Read and process data in a streaming context**.

  It has been shown that large sample streams of surveillance data can be ingested into the online platform without significant problems in terms of duplicate and out of order signals. The verification experiments were, however, performed against a cleaned data sample and further testing will be required against more raw data sources.
- **VER-OBJ-ONLINE-2.1.2 Calculate the flight efficiency indicators accurately along the flight trajectory**

  The advanced performance data model has been shown to be capable of calculating a complete set of AURORA performance indicators in near real-time without significant divergence from the indicators calculated offline. The prediction of indicators was not completed as additional algorithms were necessary to generate new full trajectories that had to comply with the requirements to run the generation and reconstruction processes, thought the consistent integration of the already flown trajectories based on ADS-B with the predicted trajectories until destination.
- **VER-OBJ-ONLINE-2.1.3 Calculate indicators efficiently enough to match specifications of AURORA STAM use cases i.e. the method to calculate off-line (trajectories generation, reconstruction and optimal routes) in combination with advanced analytics is feasible to obtain the efficiency indicators online**

  Across the verification experiments performed the average latency between a surveillance point being received into the pipeline and a new performance indicator being made available was 30 seconds. During workshops with AURORA airspace users' workshop of an upper limit on latency of 5 minutes was established and so the current solution is well within this bound. The highest throughput observed in our test datasets covering the ECAC region is approximately 350 surveillance points per second. The current platform has been shown to be more than capable of handling this throughput.
- **VER-OBJ-ONLINE-2.1.4 Select the most appropriate streaming data technology for the AURORA STAM use case**

  Rather than building complete systems using different technology options different options for various components were evaluated and the most appropriate technologies were integrated into the overall pipeline. The current platform uses standard technologies for streaming big data scenarios and has been demonstrated to meet the latency and throughput requirements of the AURORA scenario.

Currently performance indicators are calculated after the completion of a flight, but by using a data streaming technology and live flight data, the AURORA project has verified that indicators can be calculated on the fly. This dynamic calculation will improve the visibility of on-line ATM performance indicators as the calculations will be quicker and the results would be readily accessible.

4.2 Recommendations

This subsection outlines recommendations for overcoming the limitations which the existing tested implementation of stream-based data model has.

Recommendation to improve the performance of trajectory reconstruction and generation service. The usage of existing service is via a shared folder on a server hosted by Boeing using SAMBA protocol. This requires a lot of disk I/O and files I/O too, which are considered as inefficient operations in computer systems. The recommendations toward it would be providing API to call on the code level so that the stream data pipeline data could receive and process it immediately via memory rather than going through disk.

Another recommendation particularly for trajectory reconstruction service is that for each call we only need to send the latest updates rather than accumulate them first as a whole trajectory.

Recommendation to integrate data cleaning process with stream data pipeline.

Recommendation to implement and integrate independent update process of reference trajectories. Some reliable criteria are required to quickly detect whether a flight is landed, so that the redundant reference trajectories can be removed from the memory periodically to save more space.

Recommendation to implement and integrate the prediction of flight efficiency indicators' value at destination. At least 30 days data for each flight should be available which contains the reconstructed trajectory and all related reference trajectories. This data can be used as training dataset for determining parameters for predication model, for example, linear regression.

5 References

[1] AURORA D4.2, “Stream-based data model”, Edition 00.01.00, Oct 2017.

[2] AURORA D3.1, “Description of experimental plans”, Edition 00.01.00, Feb 2017.

[3] AURORA D3.2, “Report on testing of user-centric air traffic efficiency indicators and local targets”, Edition 00.01.00, July 2017.

[4] “Apache Kafka” [Online]. Available: https://kafka.apache.org/.

[5] “Apache Spark” [Online]. Available: https://spark.apache.org/.

[6] D. González-Arribas, M. Soler, J. López-Leonés, Enrique Casado and M. Sanjurjo-Rivo, ‘Automated Optimal Flight Planning based on the Aircraft Intent Description Language’, IEEE Transactions on Intelligent Transportation Systems.

[7] User Manual for the Base of Aircraft Data (BADA) Revision 3.13 - Eurocontrol Experimental Centre

[8] User Manual for the Base of Aircraft Data (BADA) Revision 4.20 - Eurocontrol Experimental Centre

[9] The GFS Atmospheric Model, Environmental Modeling Center - NOAA (August 2003) http://www.emc.ncep.noaa.gov/gmb/moorthi/gam.html

[10] NCEP/EMC GFS Model Website http://www.emc.ncep.noaa.gov/GFS/

[11]Heavenly Mathematics The Forgotten Art of Spherical Trigonometry, Glen Van Brummelen, Princton University Press, 2013.

-END OF DOCUMENT-