Western Sydney Airport

System Integration Specification Document for AOS and Sydney Metro CCS

Document Reference	Aconex Reference
	WSA60-WSA-00050-PM-SPC-000125
WBS Reference	Revision
	01
Integration Party 1 (IP1)	Integration Party 2 (IP2)
AOS	SYDNEY METRO CCS

OFFICIAL

Document Control Sheet

Project Name	Western Sydney International Airport: Sydney Metro		
Document Title	System Integration Specification Document for AOS and SYDNEY METRO		
Document No.	WSA60-WSA-00050-PM-SPC-000125		
File Name	WSA60-WSA-00050-PM-SPC-000125 SISD AOS & SYDNEY METRO.docx		
Revision	01		
Date (DD-MM-YYYY)	21-08-2025		
	Prepared By:	**Checked By:**	**Approved By**
Name	Gopal Bhumireddy	Mary Joy Sabal	Pradeep Yedatore

Revision History

Rev.	Date	Comments	Author(s)	Reviewer(s)	Approver
A	29-07-2025	Initial Version	Gopal Bhumireddy	Mary Joy Sabal / Stephen Dodds	Pradeep Yedatore
00	06-08-2025	Issued for Approval	Gopal Bhumireddy	Mary Joy Sabal	Pradeep Yedatore
01	21-08-2025	CRS Feedback	Gopal Bhumireddy	Mary Joy Sabal	Pradeep Yedatore

Document Acceptance

WSA Use Only			
Name:	**Position:**	**Signature:**	**Date:**
Mark Lownds	General Manager Qvest – Parklife Metro (mark.lownds@qvest.com)		
Abhijith Ramesh	System Manager CCS - Parklife Metro (abhijith.ramesh@siemens.com)		
Phanidhar Boddu	Customer Success Manager Amadeus (phanidhar.boddu@amadeus.com)		
Christopher Kwok	Account Executive – Western Sydney Airport DXC Technology (Christopher.kwok@dxc.com)		
Graeme Edwards	TDP Program Manager Western Sydney Airport (GEdwards@wsiairport.com.au)		
Comments:			

Table of Contents

1 Introduction

1.1 Purpose

This document describes the integration of the Airport Operational System (AOS) with Sydney Metro CCS system for the Western Sydney Airport (WSA) project.

This integration is based on the IATA PADIS specifications and is intended to exchange flight information between the Amadeus AOS and the CCS implementation utilising the WSA integration platform as a service (iPaaS) as the supporting middleware.

In particular:

- The exchange of operational flight information between AOS and Sydney Metro CCS is based on IATA's Aviation Information Data eXchange (AIDX) message version 22.1 with extension data elements to display flight details in Sydney Metro.

1.2 Target Audience

This document has been developed for WSA as a part of the Enterprise Technology Contract. The audience for this document is the following:

- WSA and DXC Enterprise / Integration Architects who are responsible for the review / delivery of Integration APIs to support the integrated set of business requirements for the WSA Airport Enterprise.

- WSA and ETC Developers who will be developing the source and target system interfaces in iPaaS.

- ETC Testers, Amadeus, and Sydney Metro Support teams

1.3 Scope

This document covers the integration between Sydney Metro CCS and AOS as described in the relevant WSA RFT document references and the Integration Register.

This document covers the following:

- Message exchanges between Sydney Metro CCS and AOS, which is automatic.
- Environment necessary to realize the integration between Sydney Metro CCS and AOS
- Communication protocols and connection handling to realize the integration between Sydney Metro CCS and AOS (utilising the iPaaS)
- Typical sequence diagrams to illustrate data flows between Sydney Metro CCS and AOS for exchanged data items.
- Exception conditions handling.
- Assumptions, Exclusions and Key Integration Decisions.

This document does not cover any system functionality or details other than those mentioned above.

1.4 Exclusions

SNO	Exclusion
1.	The detailed list of fields is documented in IDM AOS & TSS Systems [Aconex ID: WSA61-WSA-00051-IM-MOD-000001].

1.5 Definitions, Acronyms and Abbreviations

#	Definition/Acronym	Description
1.	ACK	Acknowledge/ Acknowledgement
2.	AFTN	Aeronautical Fixed Telecommunications Network
3.	AIDX	Aviation Information Data eXchange. AIDX is the IATA defined, global XML messaging standard for exchanging flight data between airlines, airports and any third party consuming the data.
4.	AOCC	Airport Operation Control Center
5.	AODB	Airport Operational Database
6.	AOS	Airport Operational System
7.	API	Application Programming Interface
8.	ATC	Air Traffic Control
9.	CCS	Central Control System
10.	DEP	Departures
11.	ESB	Enterprise Service Bus
12.	ETC	Enterprise Technology Contract
13.	FIDS	Flight Information Display System
14.	FRMS	Fatigue Risk Management System
15.	HTTP	Hypertext Transfer Protocol
16.	GA	General Aviation
17.	IATA	International Airport Transport Association
18.	IDM	Integration Data Model
19.	iPaaS	integration Platform as a Service
20.	IT	Information Technology
21.	ITSM	IT Service Management
22.	NA	Not Applicable
23.	PADIS	Passenger and Airport Data Interchange Standards.
24.	Pub-Sub	Publish Subscribe Pattern
25.	REST	Representational state transfer
26.	RFT	Request For Tender
27.	SaaS	Software as a Service
28.	SIBT	Scheduled In-Block Time
29.	SISD	System Integration Specification Document
30.	SIT	System Integration Testing
31.	SM	Sydney Metro
32.	SMF	Solace Message Format
33.	SOBT	Scheduled Off-Block Time

#	Definition/Acronym	Description
34.	Solace	iPaaS managed broker
35.	SOP	Standard operating procedures
36.	SWTC	Scope of Works and Technical Criteria
37.	TBC/ TBD	To be confirmed/ To be decided
38.	TCP/IP	Transmission Control Protocol/Internet Protocol
39.	WSI	Western Sydney Airport IATA code
40.	UML	Unified Modelling Language
41.	UTC	Universal Time Co-ordinated. UTC is the primary time standard by which the world regulates clocks and time.
42.	UUID	Universally Unique Identifier. A UUID is a randomly generated identifier that is universally unique for all practical purposes.
43.	UTF	Unicode Transformation Format
44.	WSA	Western Sydney Airport
45.	WSI	Western Sydney Airport IATA code
46.	XML	eXtensible Markup Language
47.	XSD	XML Schema Definition

1.6 References

#	Document Title	Version	Issuer	Date
1.	XML Implementation Guide Aviation Information Exchange (AIDX)	22.1	IATA, A4A & ACI	01-03-2022
2.	PADIS EDIFACT and XML codeset	19.1	IATA	01-03-2019
3.	AOS Blueprint Aconex ID: WSA60-WSA-00050-PM-SPC-000001 Architecture RevA	B	Amadeus	02-08-2023
4.	AOS RFT	00	WSA	22-03-2023
5.	iPaaS Blueprint Aconex ID: WSA61-H.P.-00050-PM-SPC-002099	F	DXC	20-02-2023
6.	WSA Future State Enterprise Architecture Aconex ID: WSA00-ACCWSA-00000-IT-GUL-000002	00	WSA	13-11-2020
7.	Integration Data Model Aconex ID: WSA61-WSA-00051-IM-MOD-000001	V06	DXC	2-07-2023
8.	INTERFACE CONTROL DOCUMENT – AOS	0.3	Amadeus	26-09-2023
9.	Integration Register Aconex ID: WSA00-ACCWSA-00000-IT-REG-000011	4.2	WSA	23-08-2023
10.	Enterprise Architecture Principles Aconex ID: WSA00-ACCWSA-00000-IT-GUL-888001	00	WSA	27-02-2023
11.	iPaaS Integration Patterns WSA61-H.P.-00050-PM-REC-000096	03	DXC	

1.7 Open items

The following table captures the open items with the owners to finalize this SISD.

SNO	Open item	Owner
1.	Sydney Metro- CCS integration with ITSM for incident creation for manual incident creation, a manual SOP needs to be established. Please refer section 2.1 item 1	WSA
2.	Sydney Metro to share Network Architecture diagram	Sydney Metro

1.8 General Assumptions

SNO	Assumption
1.	The present and expected future flight movement volumes are not expected to reach a size limit (i.e. 100MB) which would require messages to be segmented. Therefore, flight information (schedule) will be managed as a single self-contained event message.
2.	AOS sends full snapshot of flight information. AOS sends FlightNumericalID (the unique record identifier of AOS) in TPA_Extension for flight identification and to receive the same for any Incoming message to AOS. AOS also sends Arrival flight SIBT for subsystem and to receive the same in the Incoming message to AOS as it requires SIBT (Date-Time) to match Arrival flights at WSI. The same for Departure flight SOBT. This is as per Section E2.4 of AIDX XML Implementation Guide i.e., on Airport oriented industry system to use the scheduled date in place of the origin date. AOS sends the unique ID for record identification and to receive the same in the Incoming message to AOS.
3.	For the Daily Flight schedule resend request/response (used for error recovery or unhappy path) a throttling limit of **10 mins** between subsequent requests will be imposed on the AOS inbound interface. This is configurable by Amadeus and will be confirmed during SIT testing. Please refer to section 5.3.1
4.	No requirements for General Aviation flights from WSI at current stage, so technically no GA flights will be sent out from AOS.
5.	LOCAL time refers to the time in Sydney.
6.	AOS will be providing response to synchronous request via IATA_AIDX_FlightLegRS. Please refer to section 1.6 #7

SNO	Assumption
7.	Message Ordering: source system is responsible for sending the message in order and target system should be using the timestamp of the record to compare it with the record in the database or at the message level for the same message type. For clarification: A newly received message should only be compared with the most recent message of the same type for that same unique flight leg (using the LegIdentifier elements). This ensures that in the rare instance of newer messages for that same unique flight leg with an older timestamp being received by the target system, can be discarded as they are stale. However, newer messages having older timestamps for a different flight leg should not get discarded just on the basis of having an older timestamp without considering the flight leg. e.g.: Flight Update - only a newer flight update message with an older timestamp value when compared to the most recent flight update message for that same unique flight leg should be discarded by the subsystem (Sydney Metro).
8.	Sydney Metro - CCS supports OAuth as an Authentication method when making calls to IPaaS APIs (e.g. to request a Flight Schedule Resend).
9.	AOS/IPaaS will provide raw flight information (schedule and updates) to SYD Metro system. The actual business rules for public flight display will need to be agreed with WSA and configured within the target system (CCS). E.g., it may be required to not show air/ground return flights; it may be required to only display flights -x and +y hours from the current time; etc.
10.	In the event that Sydney Metro CCS is down and not connected to iPaaS, the messages received by iPaaS from AOS over that period will be discarded. Once Sydney Metro CCS reconnects to iPaaS, to recover the messages, Sydney Metro CCS would invoke a recap request to iPaaS. Please refer to section 1.6 #11

1.9 Integration Register

Interface ID	Description	Source	Target	Pattern	Data Elements
WSA_SYSIF_009_p	Flight Update	AOS	Sydney Metro	Event Message	ETA, ETD, On/Off Block, Actual Land/Take Off time, Flight Ad-hoc, Update, Delete
WSA_SYSIF_443_g	Daily Flight Schedule	AOS	Sydney Metro	Event Message	Flight Schedule

1.10 Requirements Cross-Reference

#	REQ #	Requirement Details	SISD Ref.
1.			

2 Integration Details

2.1 Key Integration Decisions

SNO	Decision	Rationale	Implication	Status/Comments
1	Sydney Metro CCS will not automatically send an alert to WSI's ITSM to log connection failures.	Sydney Metro CCS is not integrated to WSI's ITSM at this stage.	It will be a manual SOP. Hence, it won't be real-time ITSM incident creation.	Yet to review with WSA
2	Sydney Metro CCS will be logging key events related to this integration in its local log files which is managed by Sydney Metro team.	Sydney Metro CCS is not integrated to WSI's Splunk which is the centralised logging and monitoring tool.	WSI's Ops Team would reach out to Sydney Metro CCS Team in the event they require the logs for further analysis.	Validated with WSA on this assumption.
3	AOS will not send an alert to ITSM to log connection failures.	AOS SWTC does not have any integration with ITSM to provide operational faults/alerts.	Due to this, any connection error that AOS has with IPaaS, will not be reported to ITSM. Instead, AOS will report error to AOS operator to alert the WSI IT Ops team.	Validated with WSA on this assumption.
4	AOS will be logging key events related to this integration in its local log files and not support centralized logging.	AOS system does not support a centralized logging mechanism like Splunk.	AOS system's integration logs will only be available locally on their system deployment. Centralized log ingestion and analysis will not be possible.	Validated with WSA on this approach.
5	AOS RESTful endpoints are not compliant with RESTful principle for error response	AOS doesn't provide standard HTTP error codes.	iPaaS will require custom error handling for all such error scenarios and will require to parse all "HTTP 200 OK" response messages to distinguish between success and error responses.	Validated with WSA on this approach.
6	AOS will only offer their health check to WS IT OPS and will not extend this service to iPaaS.	AOS does not support a dedicated HealthCheck URL that iPaaS can use in its Message Reliability Solution.	iPaaS will not do custom health check for connectivity with AOS.	Validated with WSA on this approach.

SNO	Decision	Rationale	Implication	Status/Comments
7	For Daily Flight Schedule and Flight Update events, SydneyMetro will directly consume directly from Solace external Topic.	Sydney Metro CCS supports event driven (pub/sub) pattern.	For future requirements wherein filtering out or transforming AIDX payload (flight updates and daily flight), then Sydney Metro will be responsible for such requirement.	Validated with WSA on this approach.
8	AOS will not support Message Buffering for “medium time disconnect” scenario as required for the Reliable Message Pattern.	Limitations of the source system (AOS)	There will be no automatic recovery in case of medium time disconnects.	Validated with WSA on this approach.
9	Usage of airline and other logos	Airline (and other relevant) logo/image information will be stored locally in SYD Metro system and will not be passed over this integration. Note: Sydney Metro and TNSW have an agreement to not use the flight logos.	N/A	Validated with WSA on this approach.

2.2 Integration Overview

This section provides an overview of how the integration will be implemented. This includes communication reasons, communication states, connectivity methods, base/static data assumptions etc.

AOS and Sydney Metro CCS will communicate with each other via an integration layer provided by the integration platform as a service (iPaaS) to exchange the various business data elements required to support the respective system operations.

1. Events are delivered as XML formatted messages based on IATA's AIDX format defined by [PADIS_XSD] version 22.1 with extension data elements. The same message format could be reused to deliver different event types, across different subsystems. Example message formats are provided in a separate Integration Data Model.
2. Sydney Metro CCS will use iPaaS to retrieve events or send acknowledgment messages. An acknowledgement signifies that the Sydney Metro CCS has successfully retrieve a message from iPaaS and it has been processed.
3. AOS will use iPaaS to send and receive events or send acknowledgment messages. An acknowledgement signifies that the AOS has successfully received a message from iPaaS and it has been processed.
4. Error handling is to be performed at an application level and discussed later in this document.

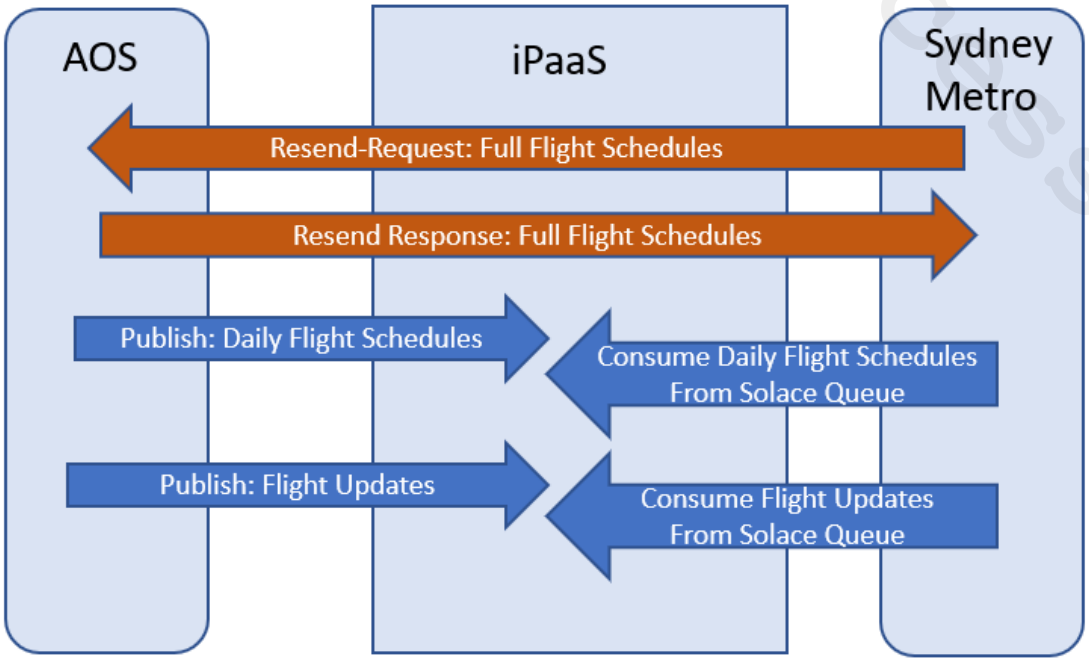


Figure 1. Integration Overview Diagram

2.3 Message Flow

This section includes UML sequence diagrams depicting the key message flows under different communication modes and states.

The following three message types shall be used to facilitate the flow of information:

- IATA_AIDX_FlightLegNotifRQ – contains one or more flight leg records.
- IATA_AIDX_FlightLegRQ – a request for flight leg records
- IATA_AIDX_FlightLegRS – a response to either of the two aforementioned RQ messages

2.3.1 Sydney Metro CCS to AOS Flow 1: Resend Request Daily Flight Schedule

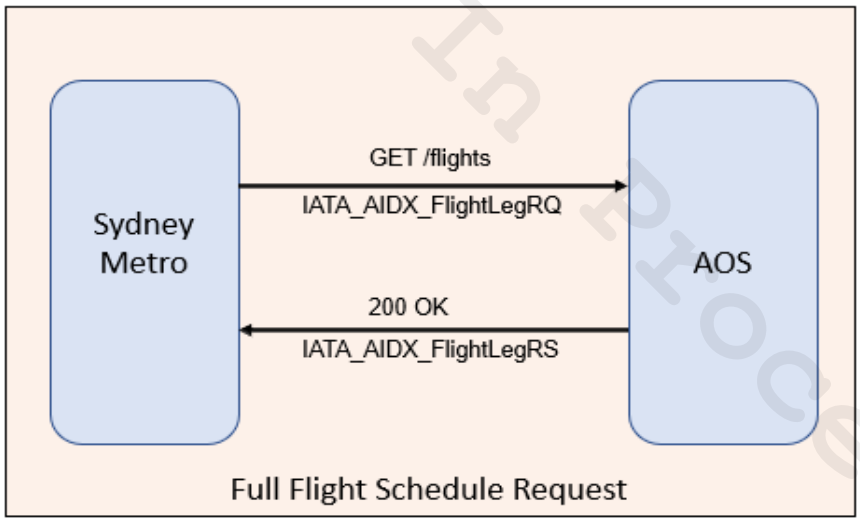


Figure 2. Flight Schedule Resend Request

2.3.2 AOS to SYDNEY METRO - CCS Flow 2: Publish Daily Flight Schedule

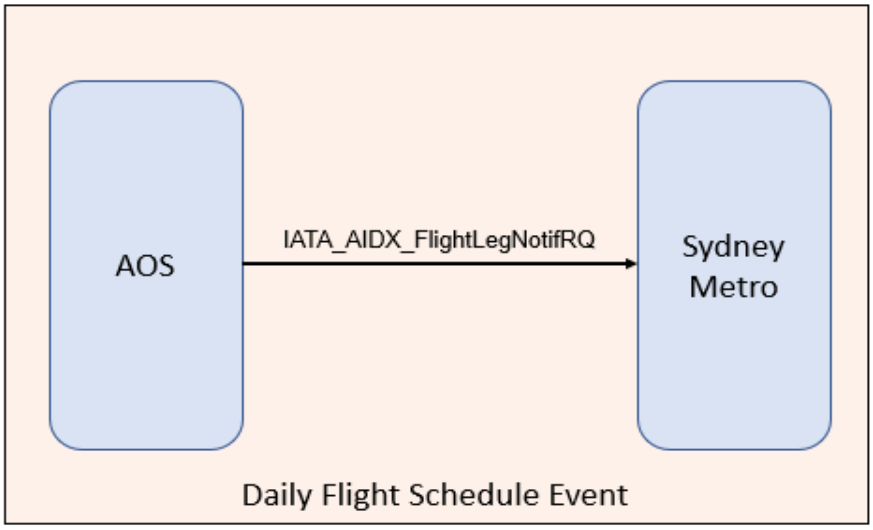


Figure 3. Daily Flight Schedule Event

2.3.3 AOS to SYDNEY METRO CCS Flow 3: Publish Flight Updates

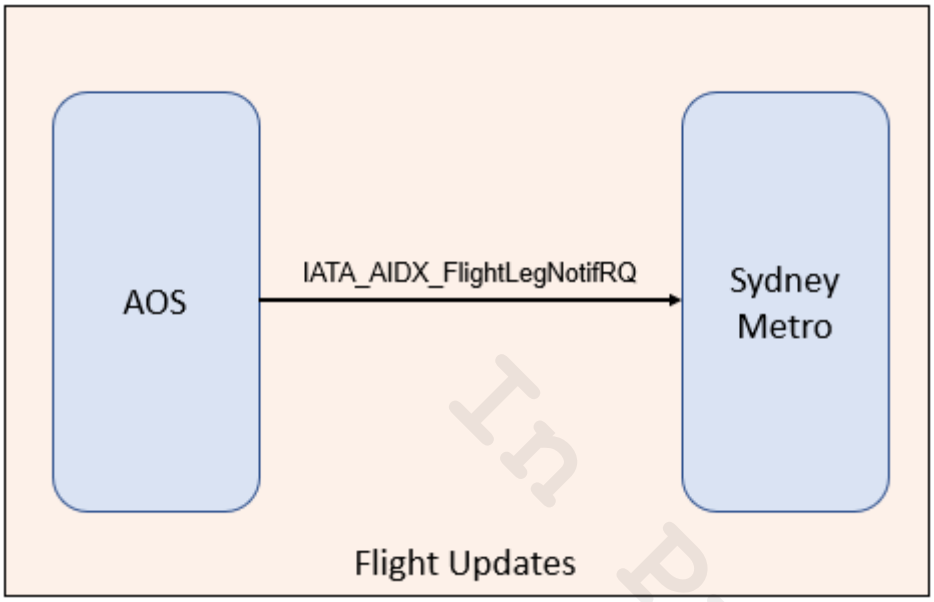


Figure 4. Flight Status Update Event

2.4 Physical Link

5. Sydney Metro and AOS will not have any direct physical link.
6. Sydney Metro and AOS will exchange XML messages based on IATA's AIDX format.
7. Please refer to AOS (WSA60-WSA-00050-PM-SPC-000001 Architecture RevA) and iPaaS (WSA61-H.P.-00050-PM-SPC-002099) blueprint documents.

2.4.1 SYDNEY METRO

Sydney Metro systems are On-prem in Sydney Metro's OCC and BOCC data centre.

OCC - Orchard Hills Stabling and Maintenance Facility Primary Technical Room

BOCC - Sydney Metro St Marys Station Telecommunication Equipment Room.

Sydney Metro will communicate to iPaaS External Solace Broker via MQTT

2.4.2 AOS

AOS is a SaaS solution hosted in Azure.

2.4.3 iPaaS

iPaaS is a SaaS solution hosted in AWS-Sydney region.

2.5 System Prerequisite Software

The tables below list all the software components required for realizing the integration between systems concerned.

2.5.1 AOS

AOS is required to perform the following initial steps to use iPaaS Restful APIs:

1. AOS needs to be onboarded to WSA Active Directory and assigned to a group.
2. Once onboarding is completed, AOS will be provided a secure login URL that uses SSO (Single Sign-On) for authentication.
3. AOS can then request access to specific RESTful APIs within Anypoint Exchange.

2.5.2 Sydney Metro

2.5.2.1 Solace

Sydney Metro is required to perform the following initial steps to use the Solace Broker MQTT

1. Sydney Metro should coordinate with WSA to procure a CA-signed certificate.
    a. The certificate must have the Common Name (CN) set to "SydneyMetro"
2. Provide iPaaS with the public certificate in PEM format to assign to the SydneyMetro Solace account.
3. We should get separate Non-Prod and Prod certificates.

2.5.2.2 MuleSoft API

Sydney Metro is required to perform the following initial steps to use iPaaS Restful APIs:

1. Sydney Metro needs to be onboarded to WSA Active Directory and assigned to a group.
2. Once onboarding is completed, Sydney Metro will be provided a secure login URL that uses SSO (Single Sign-On) for authentication.
3. Sydney Metro can then request access to specific RESTful APIs within Anypoint Exchange.

2.5.2.3 CyberArk

Sydney Metro will raise ServiceNow ITSM "Non-Standard Request" to be on-boarded into CyberArk Vault so they can retrieve iPaaS credentials.

2.6 Online Protocol

2.6.1 Security Management

2.6.1.1 AOS

Security Mechanism	Components	TEST	PROD
OAuth 2.0	Client ID: api_key	1A\\jboss/edit	AITA-WSI\ipaasac
	ClientSecret: api_secret	Retrieve from CyberArk. Platform: **Azure Portal** Safe: **SDY-IT-APP-AOS-I3** Username: 1A\\jboss/edit	Retrieve from CyberArk. Platform: **Azure Portal** Safe: **SDY-IT-APP-AOS-I3** Username: AITA-WSI\ipaasac
	Grant Type: client_credentials		
	Access Token: access_token		
	Token Type: Bearer		

Token lifecycle

Once created, the token will be valid for a certain amount of time expressed by the expires_in parameter. Before calling the service API, you can either check that the token is not expired or capture the unauthorized error (http code = 401). In both cases, you must request a new token.

2.6.1.2 Sydney Metro – CCS

Sydney Metro to use MQTT with iPaaS (Solace).

Security Mechanism	Components	TEST	PROD
MQTT	Client username	SydneyMetro	SydneyMetro
	Message VPN	WSA_EXT_TEST	WSA_EXT_PROD
	Secured MQTT URL	tcps://mr-connection-h22m13eu241.messaging.solace.cloud	tcps://mr-connection-jopyt59fu90.messaging.solace.cloud
	Secured Port	55443	55443
	Certificate	CA-Signed SH-2 PEM Format Issued by CA	CA-Signed SH-2 PEM Format Issued by CA

Sydney Metro client to ensure guaranteed messaging:

Scenario	Quality of Service (QoS)
Sydney Metro to iPaaS (Solace)	1

2.6.1.3 iPaaS

Secrets will be stored in a secure vault.

Application	Security Mechanism	Components	TEST	PROD
wsa-aos-flights-eapi	Oauth2.0	tenant	2e4116c4-b8b1-4da3-8a90-2812e2150f3d	2e4116c4-b8b1-4da3-8a90-2812e2150f3d
		Client ID	020a3ce1-3e8e-4f43-bac3-365ae0abd194	73498769-0366-4e83-b7fa-39f9a0c28920
		Client Secret	Retrieve from CyberArk. Platform: SDY-IT-APP-Integration Safe: SDY-IT-APP-AOS-I3 Username: 020a3ce1-3e8e-4f43-bac3-365ae0abd194	Retrieve from CyberArk. Platform: SDY-IT-APP-Integration Safe: SDY-IT-APP-AOS-I3 Username: 73498769-0366-4e83-b7fa-39f9a0c28920
		grant_type	client_credentials	
		scope	https://graph.microsoft.com/.default	
	XML Threat Protection			
	Message processing policy			
wsa-sydmetro-flights-eapi	Oauth2.0	tenant	2e4116c4-b8b1-4da3-8a90-2812e2150f3d	2e4116c4-b8b1-4da3-8a90-2812e2150f3d
		Client ID	TBC	TBC
		Client Secret	Retrieve from CyberArk. Platform: SDY-IT-APP-Integration Safe: TBC Username: TBC	Retrieve from CyberArk. Platform: SDY-IT-APP-Integration Safe: TBC Username: TBC
		grant_type	client_credentials	
		scope	https://graph.microsoft.com/.default	
	XML Threat Protection			
	Message processing policy			
wsa-aos-flights-sapi	Client ID enforcement	Client ID: Client Secret:		
	Message Logging Policy			
	Message Logging Policy			

Note:

Client secrets used for Oauth2.0 to access iPaaS APIs must be regenerated by IT OPS every 6 months as part of credential lifecycle management process.

See section 2.6.1.3.1.5 for client secret renewal for Oauth2.0.

2.6.1.3.1 OAuth 2.0

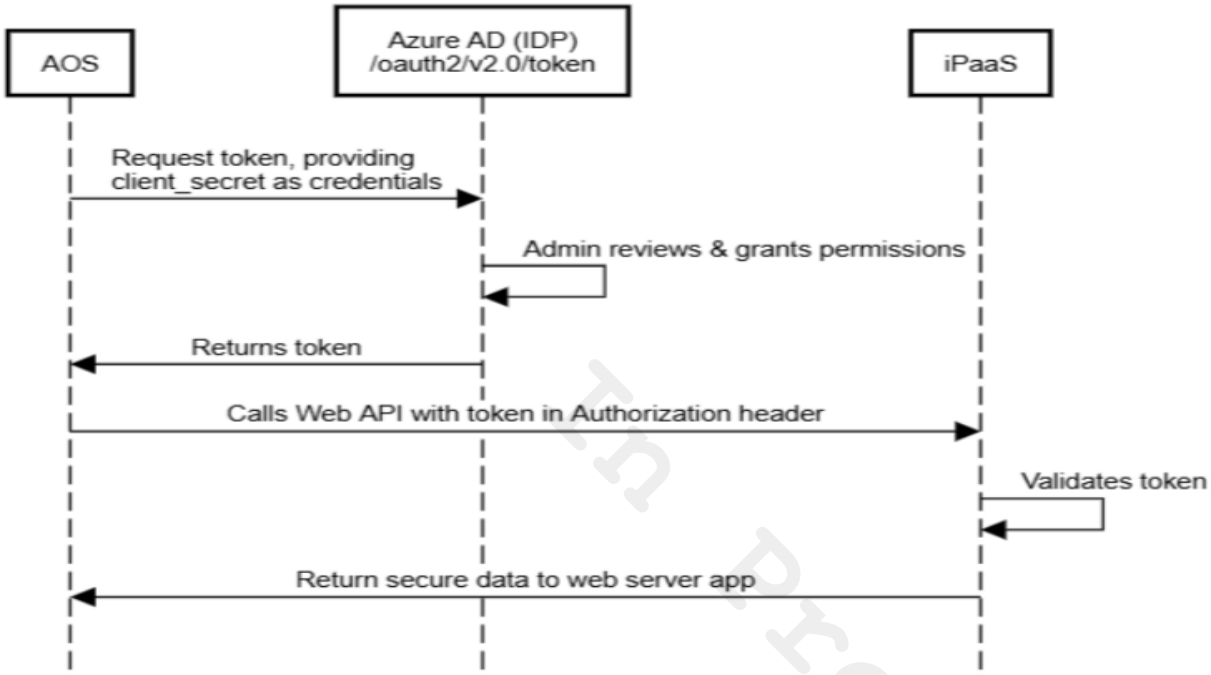


Figure 5. Oauth2.0 Client Credentials Grant Flow

2.6.1.3.1.1 Access Token Shared

Parameter	Condition	Description	Actual Value
tenant	Required	The directory tenant the application plans to operate against, in GUID or domain-name format.	2e4116c4-b8b1-4da3-8a90-2812e2150f3d
client_id	Required	The application ID that's assigned to your app. You can find this information in the portal where you registered your app.	See Section 2.6.1.2 of wsa-aos-flights-eapi
client_secret	Required	The client secret that you generated for your app in the app registration portal. The client secret must be URL-encoded before being sent. The Basic auth pattern of instead providing credentials in the Authorization header, per RFC 6749 is also supported.	See Section 2.6.1.2 of wsa-aos-flights-eapi
Grant_type	Required	Must be set to client_credentials.	Client_credentials
scope	Required	The value passed for the scope parameter in this request should be the resource identifier (application ID URI) of the resource you want, affixed with the .default suffix. All scopes included must be for a single resource. Including scopes for multiple resources will result in an error. For the Microsoft Graph example, the value is https://graph.microsoft.com/.default. This value tells the Microsoft identity platform that of all the direct application permissions you have configured for your app, the endpoint should issue a token	https://graph.microsoft.com/.default

		for the ones associated with the resource you want to use.	

2.6.1.3.1.2 Success Token Response

Parameter	Description
access_token	The requested access token. The app can use this token to authenticate to the secured resource, such as to a web API.
Token_type	Indicates the token type value. The only type that the Microsoft identity platform supports is bearer.
Expires_in	The amount of time that an access token is valid (in seconds).

2.6.1.3.1.3 Error Response

Parameter	Description
error	An error code string that you can use to classify types of errors that occur, and to react to errors.
Error_description	A specific error message that might help you identify the root cause of an authentication error.
Error_codes	A list of STS-specific error codes that might help with diagnostics.
Timestamp	The time when the error occurred.
Trace_id	A unique identifier for the request to help with diagnostics.
Correlation_id	A unique identifier for the request to help with diagnostics across components.

2.6.1.3.1.4 Use Token

Now that you've acquired a token, use the token to make requests to the resource. When the token expires, repeat the request to the /token endpoint to acquire a fresh access token.

2.6.1.3.1.5 Client Secret Renewal for Oauth2.0

1. Hosting team will need to notify the IT OPS a month before the Client secret expires.
2. IT Ops will raise a change request ticket via ServiceNow by ITSM to proceed with the changes. Both the old and new secret can work in tandem/parallel until the old secret has expired.
3. IT OPS will generate a new client secret.
4. The new secret once regenerated will be updated in the shared CyberArk safe/vault.
5. Perform a test using the new client secret.

2.6.2 Connection Management

2.6.2.1 Connection

2.6.2.1.1 AOS

AOS is offering RESTful web services for receiving messages, as such, interactions are stateless, therefore do not require an explicit connection request.

2.6.2.1.2 Sydney Metro CCS

Sydney Metro will be using MQTT to subscribe and consume from iPaaS (Solace). Message exchange will happen using external topics exposed by iPaaS (Solace) which requires a connection request. Sydney Metro must provide a valid client certificate to authenticate to Solace.

2.6.2.1.3 iPaaS

iPaaS is offering RESTful web services for receiving messages, as such, interactions are stateless, therefore do not require a connection request. For the Solace Event Broker, Client profiles will not be restricted on concurrent consumers (instances or replicas). Sydney Metro will be limited to a maximum of 250 client connections as a default.

2.6.2.2 Keep-Alive

HTTP Connection

Connection idle timeout is the number of milliseconds that a connection can remain idle before it is closed.

Response timeout is the maximum time in milliseconds that the request element blocks the execution of the flow waiting for the HTTP response.

Disclaimer:

The default value might change as it depends upon the SLA and performance requirements as well as the testing outcomes. All values must be configurable through external properties.

2.6.2.2.1 AOS

Scenario	Connection	Connection Idle Timeout	Response Timeout
AOS to iPaaS	Persistent (default)	60000ms (default)	60000ms (default)

2.6.2.2.2 Sydney Metro CCS

iPaaS will provide a Solace-based broker that will be exposed for message exchange with Sydney Metro.

Scenario	Connection	Read Timeout	Connect Timeout
Sydney Metro to iPaaS (Solace)	Persistent (default)	60000ms (default)	60000ms (default)

iPaaS will provide an API that will be exposed for message exchange with Sydney Metro.

Scenario	Connection	Connection Idle Timeout	Response Timeout
SydneyMetro to iPaaS(API)	Persistent (default)	3mins	3mins

2.6.2.2.3 iPaaS

Integration Platform provides support for Keep-Alive. It offers the flexibility to configure the default values according to the requirements.

Disclaimer:
The default value might change depends upon the SLA and performance requirements as well as the testing outcomes.

Scenario	Connection	Connection Idle Timeout	Response Timeout
iPaaS to AOS	Persistent (default)	3mins	3mins

2.6.2.3 Disconnection

It will be the responsibility of each system to disconnect gracefully from iPaaS. In case of an abnormal disconnection, the exception should be handled properly, and appropriate actions taken based on the nature of failure.

2.6.2.3.1 AOS

AOS is offering RESTful web services for receiving messages, as such, interactions stateless, therefore does not require a disconnection request.
Upon interface close-down persistent connections for keep-alive will be closed gracefully.

2.6.2.3.2 Sydney Metro CCS

Sydney Metro will use the iPaaS Solace broker for receiving messages. In the case of Sydney Metro disconnecting from Solace, the messages will be lost. Please refer to Section 1.8 - #11

2.6.2.3.3 iPaaS

iPaaS is offering RESTful web services for receiving messages, as such, interactions stateless, therefore do not require a disconnection request.

Upon interface close-down persistent connections for keep-alive will be closed gracefully.

2.6.3 Application Protocol

2.6.3.1 Message Acknowledgments

2.6.3.1.1 Pub-Sub (Asynchronous) –Flight Schedules & Updates

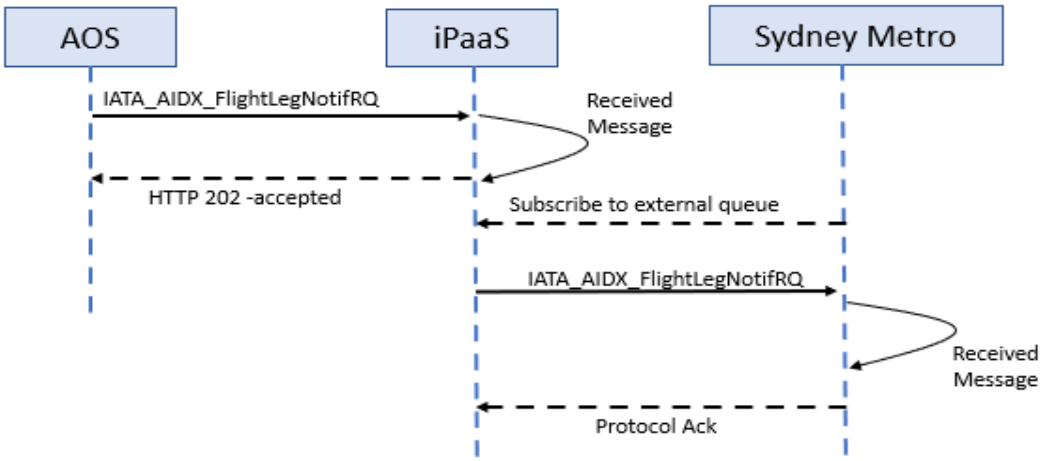


Figure 6. Acknowledgements – Flight Schedules/Updates

2.6.3.1.2 Synchronous – Resend-Request

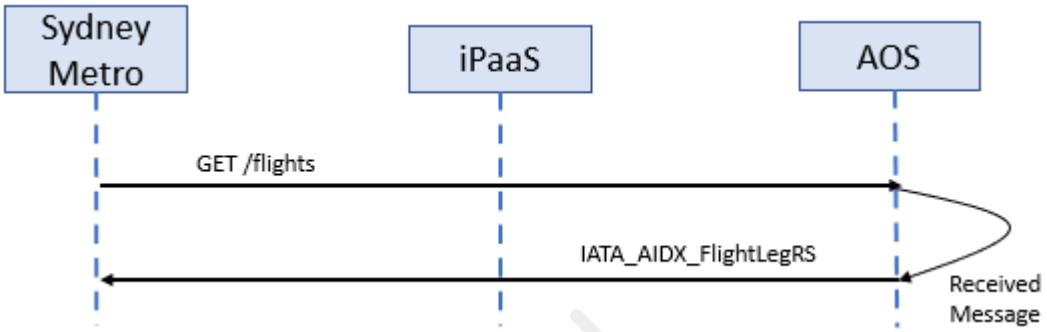


Figure 7. Acknowledgements -Resend Request

2.6.3.2 Message Processing

This section describes the typical application processing, identifying where processing starts, when processing ends and what signifies successful processing to the system connected on the other end.

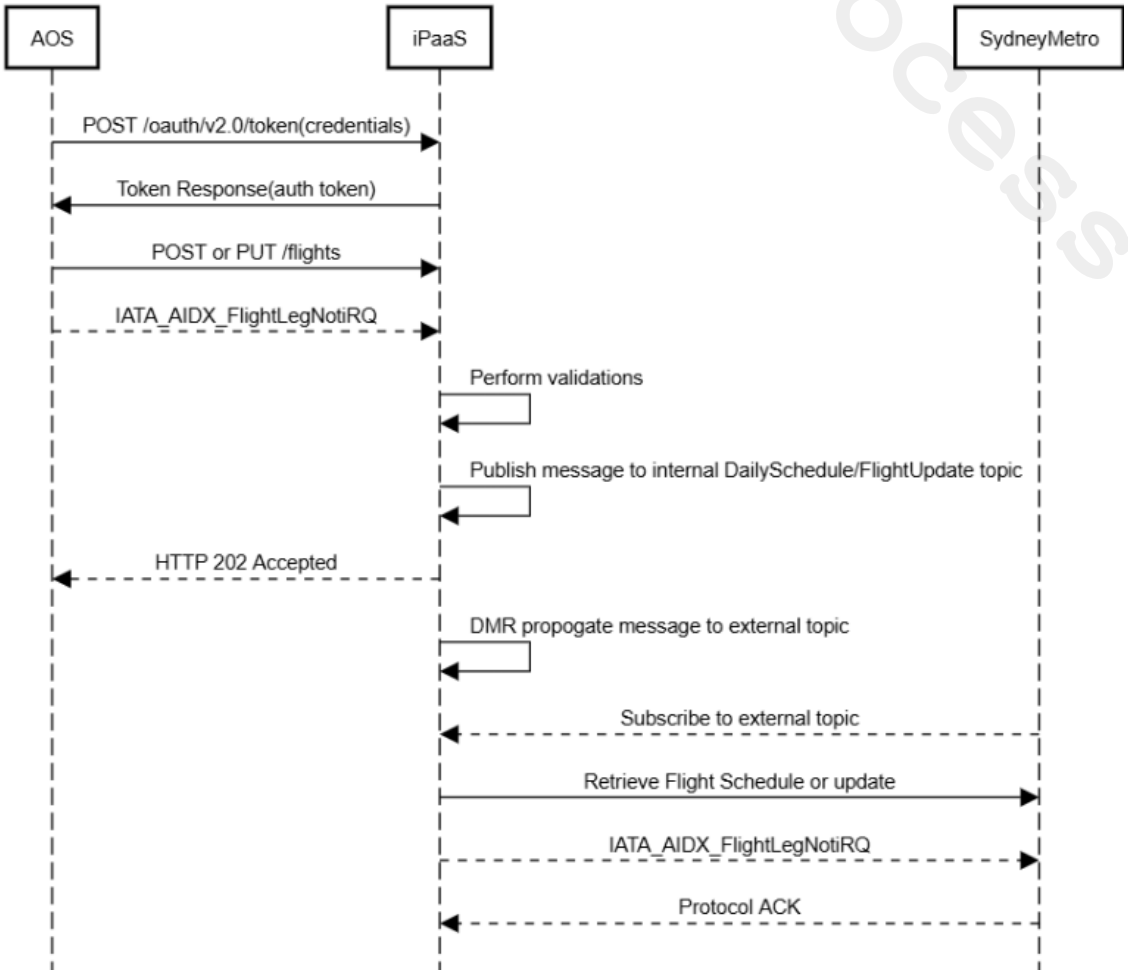


Figure 8. Flight Updates and Daily Flight Schedules

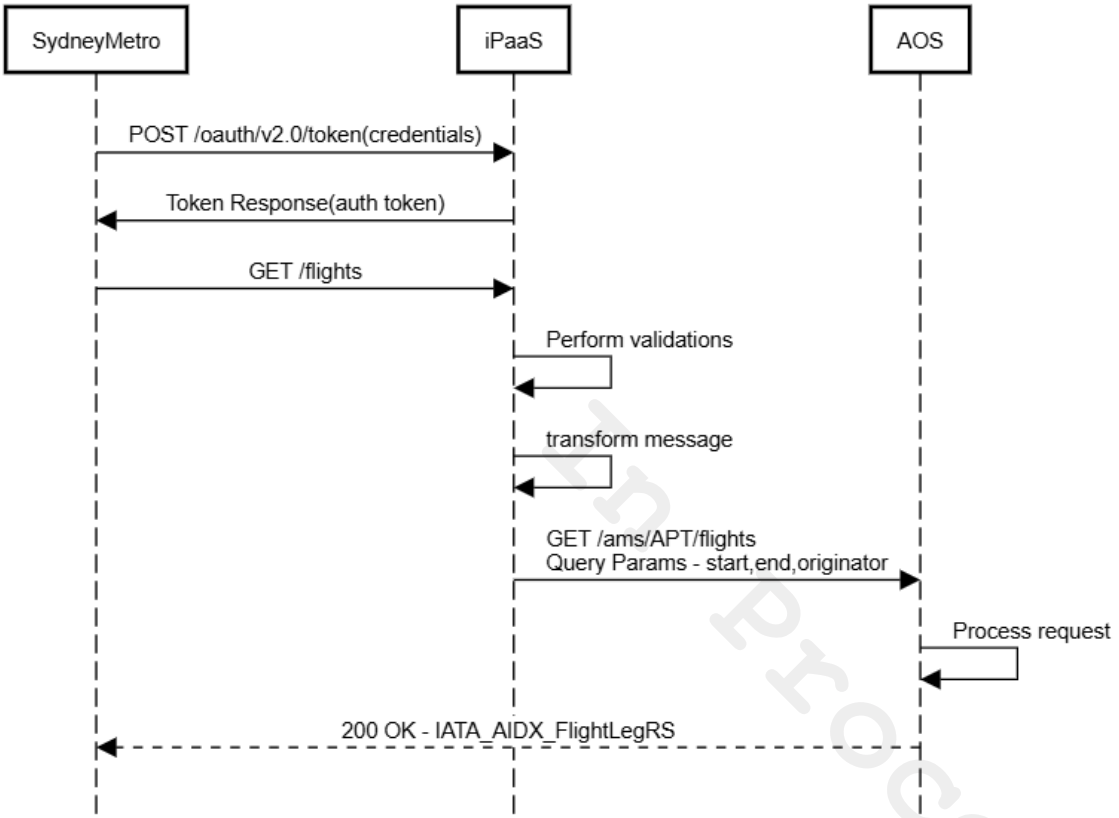


Figure 9. Resend Request-Response

2.6.3.2.1 Sydney Metro - CCS incoming message processing

Sydney Metro will subscribe to the iPaaS (Solace broker) to consume the messages published by AOS. Each message/event published by iPaaS to Sydney Metro will be received by the external topic. Upon Sydney Metro successfully consuming the message, the message listener registered with the subscriber will return a protocol level ACK to iPaaS (Solace). The message reference ID must be provided in the acknowledgement. This will ensure Solace removes the correct message from the topic.

Sydney Metro will set their message delivery mode QoS to 1 to ensure that a message is delivered to them at least once - this guarantees reliability, as the Sydney Metro will retry if it can't connect to Solace. Please see section 5.1.2.1 for Sydney Metro's reconnection strategy.

2.6.3.2.2 Sydney Metro - CCS outgoing message processing

Sydney Metro API will invoke iPaaS REST API to get full flight schedules from AOS via iPaaS. iPaaS will do necessary validations before responding back to Sydney Metro.

2.6.3.2.3 iPaaS incoming message processing

iPaaS will use an API with a REST API listener to receive messages from AOS/Sydney Metro, validate the message based on predefined rules (XML schema) if necessary, transform it into a canonical data format (if needed) before placing it on a configured TOPIC (if the message is a publish/subscribe delivery mechanism) or call target API.

2.6.3.2.4 iPaaS outgoing message processing

iPaaS contains the logic of publishing real time messages/daily flight schedules, etc from subsystems to defined topics and queues through the underlying Solace message broker software. DMR will propagate the message from the internal queue to the matching external topic for Sydney Metro to consume.

2.6.3.2.5 AOS Incoming message processing

AOS will be providing a REST API endpoint for iPaaS to post messages from iPaaS. Upon receiving the message, AOS will parse the message and update the AOS system appropriately. Upon successful message processing completion, a 200/201 success code will be returned to iPaaS.

2.6.3.2.6 AOS outgoing message processing

1. For flight schedule messages, iPaaS will be providing a REST API endpoint for AOS to post messages to Sydney Metro CCS. Upon receiving the message, the iPaaS message listener (RESTful interface) will publish the message to the internal Flight Schedule topic. Upon successful message storage on the topic, a HTTP 202 success code will be returned by iPaaS to AOS.
2. For flight update messages, iPaaS will be providing a REST API endpoint for AOS to post messages to Sydney Metro CCS. Upon receiving the message, the iPaaS message listener (RESTful interface) will publish the message to the internal Flight Update topic. Upon successful message storage on the topic, a HTTP 202 success code will be returned by iPaaS to AOS.

2.6.3.3 Message Buffering

2.6.3.3.1 AOS

AOS doesn't support Message Buffering for "medium time disconnect" scenario. iPaaS provides reliable messaging. Topics will be defined as durable and when queues are used for point-to-point communication, messages are retained for the duration of the message expiry. In the event the iPaaS goes down, these buffered messages will be retained and available when iPaaS starts up.

Refer to exception handling section for handling the situation where a subsystem endpoint is unreachable

2.6.3.3.2 Sydney Metro CCS

All durable messages from Sydney Metro CCS will be buffered.

2.7 File Protocol

Not applicable for the integration between AOS and Sydney Metro CCS. Sydney Metro CCS is not supporting importing daily schedule as a file.

2.8 Data Item Classification

2.8.1 Manual Sync Data Items

Note: Manual Sync or data via File Transfer is not applicable for this SISD.

2.8.2 Automatic Data Items

Data items listed in the table below will be synchronized automatically between and AOS i.e., changes made in the data owning system will reach the data recipient system in real time.

SNO	Type of Data	Owner	Mode of Synchronization	Frequency	Delivery Mechanism
1.	Daily Schedule	AOS	Message containing complete schedule performed automatically and sent using iPaaS messaging API	Performed automatically at 01:30 Local Time (configurable).	Pub-Sub (internal to external via DMR)
2.	Flight Update	AOS	Online XML message containing a complete snapshot of all the latest data available for this particular flight in AOS or containing incremental flight record.	Instant – triggered as/when changes occur in AOS	Pub-Sub (internal to external via DMR)
3.	Flight Daily Schedule Resend Request	Sydney Metro	Online XML message containing the daily schedule data for the requested time period	As/when Sydney Metro requires manual data synchronization	Point to point (synchronous Req/Resp)
4.	Flight Daily Schedule Resend Response	AOS	Online XML message containing complete schedule as response to request from Sydney Metro	Instant – triggered as/when requested from Sydney Metro	Point to point (synchronous Req/Resp)

2.9 Generic Message Formats

2.9.1 Date-Time Format

All date/time representations shall include the time zone indication.

[AIDX_IMP] section 3.1.10

All date and time references in AIDX must be in UTC and must be explicitly shown as UTC using a trailing Z. For example: 2012-08-25T11:35:00Z.

AOS will be using UTC time.

Sydney Metro - CCS will receive in UTC Date time from iPaaS and they should convert into to local time.

3 Sydney Metro to AOS Event Details

3.1 Daily Flight Schedule Resend Request

3.1.1 Event Attributes

Event Name	Daily Flight Schedule Resend Request
Event Description	This event occurs whenever the operator in Sydney Metro has decided to manually synchronize all flight schedule data in Sydney Metro with AOS
Integration Number	WSA_SYSIF_443_g
Single Recipient	Yes
Event Trigger	Unscheduled, triggered by Operator action in Sydney Metro. Reasons for this include: • Sydney Metro system down for a long time and needs to get the latest data from AOS. • Sydney Metro data has become corrupted and needs recovery.
Frequency	Ad hoc
Date Range	Single daily schedule request record in single XML message for the current daily schedule.
Data Count	NA
Sort Sequence	None
Acknowledgment	Application level
Priority	High
Expiry	NA
Message Fragmentation	None
Encoding	UTF 8
Time	All time fields in the message refer to UTC unless otherwise specified explicitly.
Comments	• AOS will take max 3 mins time to process Resend Request and Sydney Metro agreed to follow the same wait time. • Sydney Metro should only send this message in case of error recovery (see Event Trigger) and only one such request is accepted within a configurable time wait (refer to section 1.8 item - 4). • In the exceptional scenario when multiple subsystems fail, recover simultaneously and perform overlapping recovery requests, there might be longer wait times and additional recovery requests required from the Sydney Metro until it is able to receive the latest schedule from the AOS via iPaaS. This is because the AOS only supports one resend/recap request within

	the time window and hence rate-limiting on-behalf of the AOS for this scenario is enforced by the iPaaS for all subsystems.
Processing Summary	1. AOS will process the schedule resend request event in its entirety or reject it in its entirety. 2. When more than one parameter is specified, they will be treated with AND condition. 3. When none of the parameters are specified, AOS will return via iPaaS all of current operational day's records (as it would while generating an unsolicited flight daily schedule from AOS). 4. When only one of the parameters is specified in the parameter pair (STD or ETD), AOS will assume the logical start or end of current operational day's records for the other parameter(s).
Processing Exceptions	1. iPaaS will perform basic validations (like field size, format, existence etc) on all fields in the XML before transmitting to AOS. If validations fail, errors will be sent back to Sydney Metro to take corrective action. 2. Further validations will be performed by AOS and is expected to include: a. Date range maximum 96 hours (97 hours at the end of DST and 95 hours at the start of DST) b. Request in operational window c. No overlapping requests 3. In all other cases, if AOS cannot process/save any flight information fully or partially, errors will be recorded in the AOS. Refer to Section 5 Interface Exception Handling Policy

3.1.2 Event Message Format

Refer to the Integration Data Model for a view of the message structure(s) and example message formats.

Please refer to section 2.4 from IDM.

3.1.3 Event Message Mappings

Not applicable.

3.1.4 Event Message Data Conversions

Not applicable.

4 AOS to Sydney Metro Event Details

4.1 Daily Flight Schedule

4.1.1 Event Attributes

Event Name	Daily Flight Schedule
Event Description	This event provides receiving subsystems a record of all flights anticipated to be operating in a 96-hour window and is used for planning purposes.
Integration Number	WSA_SYSIF_443_g
Single Recipient	No, this event will be published for many systems to receive simultaneously.
Delivery Mechanism	Publish and Subscribe – Online Point to Point (Operator Initiated Error/Recovery Action)
Event Trigger	Send out by AOS automatically at 01:30 (Local Time) as a default time (configurable) or manually (error scenario) as part of Flight Schedule Resend Request
Date Range	When sent from AOS to Sydney Metro CCS (unsolicited by a subsystem), a daily schedule message will contain 00:00 D-1 to 23:59 D+2 flights (Local Time based on Flight Scheduled On-Off Block Time). Note: Date range maximum 96 hours (97 hours at the end of DST and 95 hours at the start of DST)
Data Count	Multiple daily flight records in single message. Number of records will depend on the number of daily flights operated at WSI. If the number of records is more than configured limit, the message will be split into multiple messages with start date and end date as http headers. (To be reviewed beyond the 10MAP/100MB capacity).
Sort Sequence	None
Acknowledgment	Protocol Level
Priority	Normal
Expiry	Since Sydney Metro CCS will create the durable queue dynamically upon connecting to the broker via MQTT, TTL message is not applicable. The message gets consume immediately once Sydney Metro successfully connects to iPaaS.
Message Fragmentation	None
Encoding	UTF-8
Time	All time fields in the message refer to UTC unless otherwise specified explicitly.

Comments	1. A daily schedule sent by AOS will not have flights delayed from the previous day. For example, if a flight is delayed from D-1, this will not be listed in the D+1 schedule (D to D+1). It is up to the Sydney Metro to retain the delayed flights in its database and clear out other old flights if necessary. 2. Daily schedule XML message is designed to contain a complete snapshot of all the latest data available in AOS. The optional fields will be filled depending on whether relevant data item is available in AOS or not. This is to ensure Sydney Metro can receive a complete snapshot of all flight data in AOS. *3.* Some of fields in daily schedule XML might not be used by Sydney Metro, in which case Sydney Metro can ignore such fields, though it will be sent as part of the standard XML message formats.
Processing Summary	1. iPaaS will publish the entire message from AOS (without data transformations) to the Solace Sydney Metro topic. Sydney Metro will consume/pull the message from the topic. 2. Sydney Metro will send a protocol level acknowledgement to iPaaS to indicate successful message consumption *3.* If iPaaS does not receive this acknowledgment from Sydney Metro then the message will be lost.
Processing Exceptions	1. iPaaS will perform basic validations (like field size, format, existence etc) on all fields in the XML before transmitting to Sydney Metro. For both business errors and technical errors iPaaS will – - log the error (iPaaS logging framework). - send back HTTP 202/400 to AOS depending on the layer where the validation failure (business error) occurs within iPaaS APIs. Refer section 4.1.1 for other technical error codes that iPaaS will send to AOS. - send it to the alert queue for iPaaS WSI IT Ops Team for manual intervention indicating the type of error (business or technical). - Details of re-processing will be specified in the IT Ops Manual.

4.1.2 Event Message Format

Refer to the Integration Data Model for a view of the message structure(s) and example formats.

Please refer to section 2.2 from IDM.

4.1.3 Event Message Mappings

Sydney Metro will receive the entire message sent from AOS.

4.1.4 Event Message Data Conversions

Not applicable

4.2 Flight Update

4.2.1 Event Attributes

Event Name	Flight Update
Event Description	This event occurs whenever the data related to a flight is updated in AOS or on creation of an adhoc/unscheduled flight in AOS.
Integration Number	WSA_SYSIF_009_p
Single Recipient	No, this event will be distributed to many systems simultaneously.
Event Trigger	Unscheduled, triggered by flight update or adhoc flight creation in in AOS. Reasons for flight update include: • A flight information change for a flight is received from an external Telex source like AFTN or SITA or ATC or base airline. • A flight information change for a flight is made manually in AOS (Operator intervention and a correction is applied). Reasons for adhoc flight creation include: • A new flight not in previously published daily schedule is received from an external Telex source like AFTN or SITA or ATC or base airline. • A new flight not in previously published daily schedule is created manually in AOS.
Date Range	This is sent out only if the data related to flight is changed for a flight which is scheduled for operation within the current operational window of '96' hours i.e. 00:00 D-1 to 23:59 D+2 flights (Local Time based on Flight Scheduled On-Off Block Time). Note: Date range maximum 96 hours (97 hours at the end of DST and 95 hours at the start of DST)
Data Count	The latest flight snapshot for a single flight within a single XML message.
Sort Sequence	None
Acknowledgment	Protocol Level
Priority	High
Expiry	Since Sydney Metro CCS will create the durable queue dynamically upon connecting to the broker via MQTT, TTL message is not applicable. The message gets consume immediately once Sydney Metro successfully connects to iPaaS.
Message Fragmentation	None
Encoding	UTF 8
Time	All time fields in the message refer to UTC unless otherwise specified explicitly.

Comments	1. Flight Update flight XML message is designed to contain a complete snapshot of all the latest data available in AOS. So, the optional fields will be filled depending on whether relevant data item is available in AOS or not. This is to ensure Sydney Metro can get a complete snapshot of all flight data in AOS. 2. Some of fields in Update flight schedule XML might not be used by Sydney Metro, in which case Sydney Metro can ignore such fields, though it will be sent as part of the standard XML message formats. 3. Flight update will show latest data from AOS for a specific flight.
Processing Summary	1. Whenever Sydney Metro receives this event from AOS, Sydney Metro will treat such an event as the best and most recent available information for the flight and replace its own corresponding local data items with those received from. 2. iPaaS will publish the entire message from AOS (without data transformations) to the Sydney Metro Solace topic. Sydney Metro will consume the message from the topic. 3. Sydney Metro will send a protocol level acknowledgment to iPaaS to indicate successful message consumption. 4. If iPaaS does not receive this acknowledgment from Sydney Metro the message will remain on the topic until the TTL has expired.
Processing Exceptions	1. iPaaS will perform basic validations (like field size, format, existence etc) on all fields in the XML before transmitting to Sydney Metro. For both business errors and technical errors iPaaS will - log the error (iPaaS logging framework). - send back HTTP 202/400 to AOS depending on the layer where the validation failure (business error) occurs within iPaaS APIs. Refer section 4.2.1 for other technical error codes that iPaaS will send to AOS. - send it to the alert queue for iPaaS WSI IT Ops Team for manual intervention indicating the type of error (business or technical). - Details of re-processing will be specified in the IT Ops Manual.

4.2.2 Event Message Format

Refer to the Integration Data Model for a view of the message structure(s) and example formats and refer to section 2.1.1 from IDM.

4.2.3 Event Message Mappings

Sydney Metro will receive the entire message sent from AOS.

4.2.4 Event Message Data Conversions

Not Applicable

4.3 Flight Daily Schedule Resend Response

4.3.1 Event Attributes

Event Name	Flight Daily Schedule Resend Response
Event Description	This event occurs in AOS as an automatic response to Sydney Metro Flight Daily Schedule Request
Integration Number	WSA_SYSIF_443_b
Single Recipient	Yes
Event Trigger	This event occurs in AOS as an automatic response to Sydney Metro Flight Daily Schedule Request
Date Range	Depends on parameters of Sydney Metro Flight Daily Schedule Request Also see AOS Flight Daily Schedule description
Data Count	Refer to 4.1.1 Daily Flight Schedule event attributes
Sort Sequence	None
Acknowledgment	Protocol level
Priority	High
Expiry	NA
Message Fragmentation	None
Encoding	UTF 8
Time	All time fields in the message refer to UTC unless otherwise specified explicitly.
Comments	Refer to Section 4.1.1- Daily Flight Schedule
Processing Summary	Refer to Section 4.1.1- Daily Flight Schedule
Processing Exceptions	Refer to Section 4.1.1- Daily Flight Schedule

4.3.2 Event Message Format

Refer to the Integration Data Model for a view of the message structure(s) and example formats and refer to section 2.3.1 from IDM.

4.3.3 Event Message Mappings

Not Applicable

4.3.4 Event Message Data Conversions

Not Applicable

5 Integration Exception Handling Policy

5.1 Sydney Metro - CCS

This section describes in detail the exception handling mechanism to be followed by under various situations in the system-to-system message exchange.

5.1.1 Application Errors

SNO	Error Code	Error Source	Error Type	Error Description
1.	400	iPaaS	Business	Bad Request: An error in the client request (Mostly due to validations)
2.	401	iPaaS	Technical	Unauthorized: The user can't be authenticated
3.	403	iPaaS	Technical	Forbidden: The server cannot give access to the resource.
4.	404	iPaaS	Technical	Not Found: The resource defined in the URL doesn't exist.
5.	405	iPaaS	Technical	Method not allowed: The resource defined does not support the requested method (e.g., GET, POST, PUT, PATCH, DELETE etc.)
6.	408	iPaaS	Technical	Request Timeout Error
7.	409	iPaaS	Technical	Conflict: A request conflict with the current state of the target resource
8.	412	iPaaS	Business	Precondition Failed: One of the validations in the request failed (Sometimes used instead of 400).
9.	413	iPaaS	Technical	Payload too large: in some cases, we might need to limit how large the request payload(body)
10.	415	iPaaS	Technical	Unsupported Media Type: the payload format is in an unsupported format.
11.	429	iPaaS	Technical	Too Many Requests: when an API rate limiting is implemented
12.	500	iPaaS	Technical	Internal Server Error: The server encountered an unexpected condition.
13.	502	iPaaS	Technical	Bad Gateway: The server, while acting as a gateway or proxy, received an invalid response from an inbound server.
14.	503	iPaaS	Technical	Service Unavailable: service is temporarily unable to handle requests due to some issues or maintenance.
15.	504	iPaaS	Technical	Gateway Timeout: The server tried to access an upstream service, and it took longer than expected

5.1.2 Connection Failures

This section describes the application-level failure handling mechanisms to be adopted when **Sydney Metro CCS** as a source is having connectivity issues with **iPaaS** for a given period.

5.1.2.1 Sydney Metro to iPaaS(Solace)

SNO	Failure Type	Handling Description
1	Connectivity error	If Sydney Metro fails to connect to iPaaS (Solace), Sydney Metro will perform standard retries i) Retry Attempts: 5 ii) Frequency (ms): 6000 The number of retry attempts and frequency are configurable on the Sydney Metro side. This will be subject for iPaaS assessment and Change Request may be required, as it may have a potential impact on iPaaS design. Once the retries are exhausted, CCS will notify the WSI SD operator.

5.1.2.2 Sydney Metro to iPaaS(API)

SNO	Failure Type	Handling Description
1	Short time disconnect (less than a few minutes)	Standard Retry Strategy should be implemented in a subsystem including configurable values with defaults for: i) Retry Attempts: 5 ii) Frequency (ms): 6000 Above values will be configurable properties since they are subject to change once a baseline has been established. Messages should not be lost since we require guaranteed delivery. For request-response style messages after all the reconnection attempts are exhausted an HTTP 504 error message should be returned. A health check endpoint will be exposed by iPaaS for Sydney Metro to perform further connectivity check.

Note: Sydney Metro CCS will not send alert to WSI's ITSM to log on connection failure. Refer to item 1 from Key Integration Decision table.

5.1.3 System Operations Logging/Tracing Policy

Sydney Metro CCS will be logging key events in local log files which will allow troubleshooting and tracing of integration messages.

Please refer to item 2 from 2.1 Key Integration Decisions.

5.2 AOS

5.2.1 Application Errors

The following table captures, how AOS will handle the various application error situations.

SNO	Error Code	Error Source	Error Type	Error Description
1.	400	iPaaS	Business	Bad Request: An error in the client request (Mostly due to validations)
2.	401	iPaaS	Technical	Unauthorized: The user can't be authenticated

3.	403	iPaaS	Technical	Forbidden: The server cannot give access to the resource.
4.	404	iPaaS	Technical	Not Found: The resource defined in the URL doesn't exist.
5.	405	iPaaS	Technical	Method not allowed: The resource defined does not support the requested method (e.g., GET, POST, PUT, PATCH, DELETE etc.)
6.	408	iPaaS	Technical	Request Timeout Error
7.	409	iPaaS	Technical	Conflict: A request conflict with the current state of the target resource
8.	412	iPaaS	Business	Precondition Failed: One of the validations in the request failed (Sometimes used instead of 400).
9.	413	iPaaS	Technical	Payload too large: in some cases, we might need to limit how large the request payload(body)
10.	415	iPaaS	Technical	Unsupported Media Type : the payload format is in an unsupported format.
11.	429	iPaaS	Technical	Too Many Requests: when an API rate limiting is implemented
12.	500	iPaaS	Technical	Internal Server Error: The server encountered an unexpected condition.
13.	502	iPaaS	Technical	Bad Gateway: The server, while acting as a gateway or proxy, received an invalid response from an inbound server.
14.	503	iPaaS	Technical	Service Unavailable: service is temporarily unable to handle requests due to some issues or maintenance.
15.	504	iPaaS	Technical	Gateway Timeout: The server tried to access an upstream service and it took longer than expected

5.2.2 Connection Failures

This section describes the application-level failure handling mechanisms to be adopted when **AOS as a source** is having connectivity issues with **iPaaS** for a given period.

SNO	Failure Type	Handling Description
1	Short time disconnect (less than a few minutes)	Standard Retry Strategy should be implemented in a subsystem including configurable values with defaults for: i) Retry Attempts: 30 ii) Frequency (ms): 10000 Above values will be configurable properties since they are subject to change once a baseline has been established. Messages should not be lost since we require guaranteed delivery. For request-response style messages after all the reconnection attempts are exhausted an HTTP 504 error message should be returned.
2	Medium time disconnect	**Not Applicable** as AOS doesn't support.

SNO	Failure Type	Handling Description
3	Long-time disconnect (over 5 mins)	A health check URL would be exposed by AOS to the WSI monitoring system to identify and log the change in status. If the integration is declared to be down (iPaaS is down), a manual re-synchronization is required from the source to target system triggered by the WSI IT Ops Team and does not require a UI. Details of this Manual IT SOP will be specified in the operations manual by the WSI IT Ops Team. This is not applicable for request-response (synchronous) message handling.

Note: AOS does not support “medium-time disconnect” scenario. Please refer to item 8 from 2.1 Key Integration Decision.

5.2.3 System Operations Logging Policy

AOS ESB component process-log is one of the Amadeus AOS component process-logs. These are used as part of the functionality in the AOS Applications, Health Status and Monitoring Tools. No provision is made for reporting to ITSM.

Please refer to item 4 from 2.1 Key Integration Decisions.

5.3 iPaaS

5.3.1 Application Errors

The following table captures, how iPaaS will handle the various application error situations.

SNO	Error Code	Error Source	Error Type	Handling Description
1.	200	AOS	Technical or Business	AOS will return this code for specific error scenarios for a set of Short Texts and Codes as per their Technical Design
2.	401	AOS	Technical	Unauthorized: Credentials you used are invalid for some reason to get session cookie.
3.	403	AOS	Technical	Forbidden: The server understands the request but refuses to authorize for example insufficient rights to a resource or invalid/expired session cookie.
4.	404	AOS	Technical	Not Found: Server could not be reached, or service not found.
5.	503	AOS	Technical	Service Unavailable: Server not able to handle the request.

For AOS Error response messages (Resend-Request only), iPaaS will send error code as per column “Error Code to Sydney Metro” as per the below table:

SNO	Error Code	Error Source	Error Type	Handling Description	Error Code to Sydney Metro	Throttling Applied
1.	493	AOS	Technical	Timeout occurred. Please retry.	503	No
2.	400	AOS	Business	Invalid message received.	400	No
3.	490	AOS	Business	Invalid date range	400	Yes
4.	429	AOS	Business	Too many requests.	429	NA

5.	413	AOS	Business	Too many flights in the date range. Please request in smaller windows.	400	Yes
6.	491	AOS	Technical	Failed to fetch data. Please retry after some time.	500	Yes

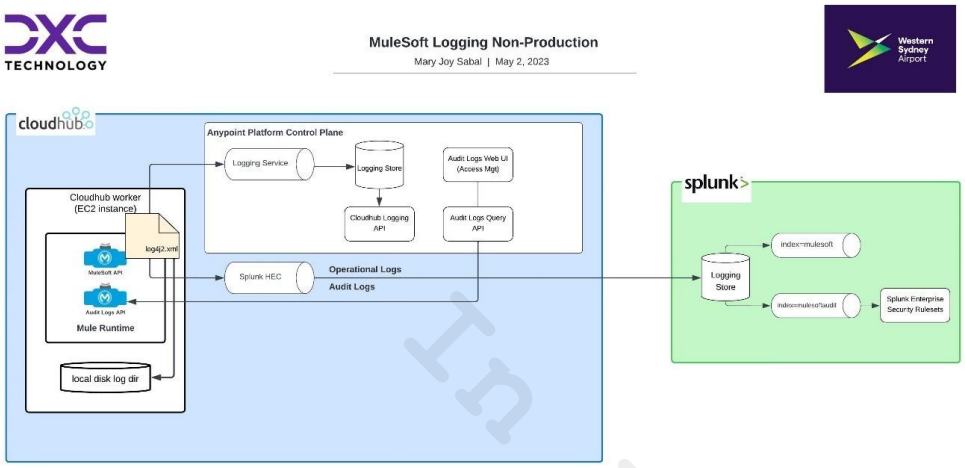
Note: The throttling of “flight schedule resend” request at AOS applies only for valid requests received and processed (parsed) successfully by AOS ESB.

5.3.2 Connection Failures

The following table have application-level failure handling mechanisms to be adopted when iPaaS side fails over or is temporarily disconnected from the Integration Platform for a given period.

SNO	Failure Type	Handling Description
1	Short time disconnect	An alert should be sent to ITSM to log this connection failure. Standard Reconnection Strategy should be implemented in a subsystem including configurable values with defaults for: i) Reconnection Attempts (Retry Count): 5 ii) Frequency (ms): 6000 Above values will be configurable properties since they are subject to change once a baseline has been established. Messages should not be lost since we require guaranteed delivery. For request-response style messages after all the reconnection attempts are exhausted an HTTP 504 error message should be returned. An alert should be sent to ITSM to log this connection failure.

5.3.3 System Operations Logging Policy



Sensitive information should never be logged to the log files. Even if it is logged in DEBUG mode, it is still a security risk, as the sensitive information would be visible by just changing the logging levels. Masking or hiding sensitive fields should be done before logging.

Sequence	Field	Description
1	correlationId	Correlation ID passed from calling application.
2	message	Placeholder for any custom message.
3	tracePoint	START (default) FLOW END AFTER_REQUEST AFTER_TRANSFORM BEFORE_REQUEST BEFORE_TRANSFORM EXCEPTION
4	originApplication	If known, application name of calling application or the queue name that the current message was pulled. Could be a queue or topic.
5	destinationApplication	If known, application name of target application or the queue or topic name that the message will be pushed. Could be a queue or topic.
6	methodName	HTTP method name.
7	elapsed	Elapsed time.
8	timestamp	Timestamp in which log was made (can be omitted if using log4j).
9	environment	Mule environment e.g., DEV, TEST, UAT, STAG, PROD

6 Resources

Note: Resource paths, Queue and Topic names should be configurable across all systems.

6.1 AOS

SNO	Event Name	Resources (includes method, URI)	Query Parameters	Headers	IDM Reference
1.	AOS OAuth URL	https://wsi.apit.amadeus.com /authentication/oauth2/token/	NA	Client ID/Client Secret	Not applicable
2.	Flight Schedule Resend Request	GET https://wsi.apit.amadeus.com/esb4/osgi/cxf/ams/APT/flights	start, end, originator	Authorization ,breadcrumbid	2.4 Flight Daily Schedule Resend Request

6.2 iPaaS

6.2.1 REST Endpoints

SNO	Event Name	API Name	Resources (includes method, URI)	Query Parameters	Headers	IDM Reference
1.	Flight Schedule Resend Request	wsa-sydmetro-flights-eapi	GET /flights	NA	Authorization	2.4 Daily Flight Schedule Resend Request
2.	Health Check Endpoint	wsa-sydmetro-flights-eapi	GET /health	NA	Authorization	NA
3.	Flight Schedule Publish	wsa-aos-flights-eapi	POST /flights	NA	Authorization	2.2 Daily Flight Schedule Publish
4.	Flight Update	wsa-aos-flights-eapi	PUT /flights	NA	Authorization	2.1 Flight Update, 2.5 Flight Delete
5.	Health Check Endpoint	wsa-aos-flights-eapi	GET /health	NA	Authorization	NA
6.	iPaaS OAuth2.0 Authentication	wsa-aos-flights-eapi	POST https://login.microsoftonline.com/ 2e4116c4-b8b1-4da3-8a90-2812e2150f3d/OAuth2/v2.0/token	NA	NA	NA
		wsa-sydmetro-flights-eapi	POST https://login.microsoftonline.com/ 2e4116c4-b8b1-4da3-8a90-2812e2150f3d/OAuth2/v2.0/token	NA	NA	NA

6.2.2 Event Broker

6.2.2.1 External Topic

SNO	Event Name	Topic	IDM Reference
1.	Flight Schedule Publish	wsa/aofm/aodb/flight/22.1/IATA_AIDX_FlightLegNotifRQ/DailySchedule	2.2 Daily Flight Schedule Publish
2.	Flight Update	wsa/aofm/aodb/flight/22.1/IATA_AIDX_FlightLegNotifRQ/FlightUpdates	2.1 Flight Update, 2.5 Flight Delete

6.3 Sydney Metro

Sydney Metro CCS will use iPaaS Platform (Solace) via MQTT topic to receive events. Refer to Section 6.2.2.2

7 Appendix

7.1 Revision 01 – Change log

Section	Comments
Document Acceptance	PLM and Qvest are added for signatories
1.7	Item 2 added
1.8	Old item 10 added into section 2.1 item 9
2.1	Item 9 added
2.6.2.2.2	Both table contents are updated
4.1.1	Processing Summary – item 3 updated.
5.3.1	Throttling Applied - column added into Resend-Request error response codes table and note added.

Certificate Of Completion

Envelope Id: EA2F4CB0-09AF-45AD-84D5-33A69E4EE072
Status: Sent
Subject: TDP Package 1 | WSA60-WSA-00050-PM-SPC-000125[01] SISD AOS & Sydney Metro
Procurement / Contract ID:
Source Envelope:
Document Pages: 45
Signatures: 0
Envelope Originator:
Certificate Pages: 4
Initials: 0
Tatiana Damianuc
AutoNav: Enabled
Level 3
EnvelopeId Stamping: Enabled
45-47 Scott St
Time Zone: (UTC+10:00) Canberra, Melbourne, Sydney
Liverpool, NSW 2170
tdamianuc@wsiairport.com.au
IP Address: 103.75.7.138

Record Tracking

Status: Original
9/9/2025 9:50:43 AM
Holder: Tatiana Damianuc
tdamianuc@wsiairport.com.au
Location: DocuSign

Signer Events	Signature	Timestamp
Mark Lownds mark.lownds@qvest.com General Manager Security Level: Email, Account Authentication (None) **Electronic Record and Signature Disclosure:** Accepted: 9/15/2025 4:52:30 PM ID: 89c7cbce-8740-4d74-a452-0d8e84a13c8a Company Name: Western Sydney Airport		Sent: 9/9/2025 9:54:20 AM Viewed: 9/15/2025 4:52:30 PM
Abhijith Ramesh abhijith.ramesh@siemens.com Security Level: Email, Account Authentication (None) **Electronic Record and Signature Disclosure:** Not Offered via Docusign		
Phanidhar Boddu phanidhar.boddu@amadeus.com Security Level: Email, Account Authentication (None) **Electronic Record and Signature Disclosure:** Not Offered via Docusign		
Chris Kwok christopher.kwok@dxc.com Security Level: Email, Account Authentication (None) **Electronic Record and Signature Disclosure:** Accepted: 7/25/2025 4:27:19 PM ID: 5a0c7cfd-1b21-4096-bdcc-f97b907bb1e4 Company Name: Western Sydney Airport		
Graeme Edwards gedwards@wsiairport.com.au Security Level: Email, Account Authentication (None) **Electronic Record and Signature Disclosure:** Not Offered via Docusign		

In Person Signer Events	Signature	Timestamp

Editor Delivery Events	Status	Timestamp

Agent Delivery Events	Status	Timestamp

Intermediary Delivery Events	Status	Timestamp

Certified Delivery Events	Status	Timestamp

Carbon Copy Events	Status	Timestamp

Alroy Rebello

arebello@wsiairport.com.au

Security Level: Email, Account Authentication (None)

Electronic Record and Signature Disclosure:
Not Offered via Docusign

Tatiana Damianuc

tdamianuc@wsiairport.com.au

Security Level: Email, Account Authentication (None)

Electronic Record and Signature Disclosure:
Not Offered via Docusign

Witness Events	Signature	Timestamp

Notary Events	Signature	Timestamp

Envelope Summary Events	Status	Timestamps
Envelope Sent	Hashed/Encrypted	9/9/2025 9:54:20 AM
Envelope Updated	Security Checked	9/15/2025 10:34:08 AM
Envelope Updated	Security Checked	9/15/2025 10:34:08 AM
Envelope Updated	Security Checked	9/15/2025 10:34:08 AM
Envelope Updated	Security Checked	9/15/2025 10:34:08 AM
Envelope Updated	Security Checked	9/15/2025 10:34:08 AM
Envelope Updated	Security Checked	9/15/2025 10:34:09 AM
Envelope Updated	Security Checked	9/15/2025 10:34:09 AM
Envelope Updated	Security Checked	9/15/2025 10:34:09 AM

Payment Events	Status	Timestamps

Electronic Record and Signature Disclosure

ELECTRONIC RECORD AND SIGNATURE DISCLOSURE

From time to time, Western Sydney Airport (we, us or Company) may be required by law to provide to you certain written notices or disclosures. Described below are the terms and conditions for providing to you such notices and disclosures electronically through the DocuSign system. Please read the information below carefully and thoroughly, and if you can access this information electronically to your satisfaction and agree to this Electronic Record and Signature Disclosure (ERSD), please confirm your agreement by selecting the check-box next to 'I agree to use electronic records and signatures' before clicking 'CONTINUE' within the DocuSign system.

Getting paper copies

At any time, you may request from us a paper copy of any record provided or made available electronically to you by us. You will have the ability to download and print documents we send to you through the DocuSign system during and immediately after the signing session and, if you elect to create a DocuSign account, you may access the documents for a limited period of time (usually 30 days) after such documents are first sent to you. After such time, if you wish for us to send you paper copies of any such documents from our office to you, you will be charged a $0.00 per-page fee. You may request delivery of such paper copies from us by following the procedure described below.

Withdrawing your consent

If you decide to receive notices and disclosures from us electronically, you may at any time change your mind and tell us that thereafter you want to receive required notices and disclosures only in paper format. How you must inform us of your decision to receive future notices and disclosure in paper format and withdraw your consent to receive notices and disclosures electronically is described below.

Consequences of changing your mind

If you elect to receive required notices and disclosures only in paper format, it will slow the speed at which we can complete certain steps in transactions with you and delivering services to you because we will need first to send the required notices or disclosures to you in paper format, and then wait until we receive back from you your acknowledgment of your receipt of such paper notices or disclosures. Further, you will no longer be able to use the DocuSign system to receive required notices and consents electronically from us or to sign electronically documents from us.

All notices and disclosures will be sent to you electronically

Unless you tell us otherwise in accordance with the procedures described herein, we will provide electronically to you through the DocuSign system all required notices, disclosures, authorizations, acknowledgements, and other documents that are required to be provided or made available to you during the course of our relationship with you. To reduce the chance of you inadvertently not receiving any notice or disclosure, we prefer to provide all of the required notices and disclosures to you by the same method and to the same address that you have given us. Thus, you can receive all the disclosures and notices electronically or in paper format through the paper mail delivery system. If you do not agree with this process, please let us know as described below. Please also see the paragraph immediately above that describes the consequences of your electing not to receive delivery of the notices and disclosures electronically from us.

How to contact Western Sydney Airport:

You may contact us to let us know of your changes as to how we may contact you electronically, to request paper copies of certain information from us, and to withdraw your prior consent to receive notices and disclosures electronically as follows:
To contact us by email send messages to: procurement@wsaco.com.au
To contact us by paper mail, please send correspondence to:
Western Sydney Airport
PO Box 397
Liverpool, NSW, 1871

To advise Western Sydney Airport of your new email address

To let us know of a change in your email address where we should send notices and disclosures electronically to you, you must send an email message to us at procurement@wsiairport.com.au and in the body of such request you must state: your previous email address, your new email address. We do not require any other information from you to change your email address.

If you created a DocuSign account, you may update it with your new email address through your account preferences.

To request paper copies from Western Sydney Airport

To request delivery from us of paper copies of the notices and disclosures previously provided by us to you electronically, you must send us an email to notices@wsiairport.com.au and in the body of such request you must state your email address, full name, mailing address, and telephone number. We will bill you for any fees at that time, if any.

To withdraw your consent with Western Sydney Airport

To inform us that you no longer wish to receive future notices and disclosures in electronic format you may:

i. decline to sign a document from within your signing session, and on the subsequent page, select the check-box indicating you wish to withdraw your consent, or you may;

ii. send us an email to procurement@wsaco.com.au and in the body of such request you must state your email, full name, mailing address, and telephone number. We do not need any other information from you to withdraw consent.. The consequences of your withdrawing consent for online documents will be that transactions may take a longer time to process..

Required hardware and software

The minimum system requirements for using the DocuSign system may change over time. The current system requirements are found here: https://support.docusign.com/guides/signer-guide-signing-system-requirements.

Acknowledging your access and consent to receive and sign documents electronically

To confirm to us that you can access this information electronically, which will be similar to other electronic notices and disclosures that we will provide to you, please confirm that you have read this ERSD, and (i) that you are able to print on paper or electronically save this ERSD for your future reference and access; or (ii) that you are able to email this ERSD to an email address where you will be able to print on paper or save it for your future reference and access. Further, if you consent to receiving notices and disclosures exclusively in electronic format as described herein, then select the check-box next to 'I agree to use electronic records and signatures' before clicking 'CONTINUE' within the DocuSign system.

By selecting the check-box next to 'I agree to use electronic records and signatures', you confirm that:

- You can access and read this Electronic Record and Signature Disclosure; and
- You can print on paper this Electronic Record and Signature Disclosure, or save or send this Electronic Record and Disclosure to a location where you can print it, for future reference and access; and
- Until or unless you notify Western Sydney Airport as described above, you consent to receive exclusively through electronic means all notices, disclosures, authorizations, acknowledgements, and other documents that are required to be provided or made available to you by Western Sydney Airport during the course of your relationship with Western Sydney Airport.