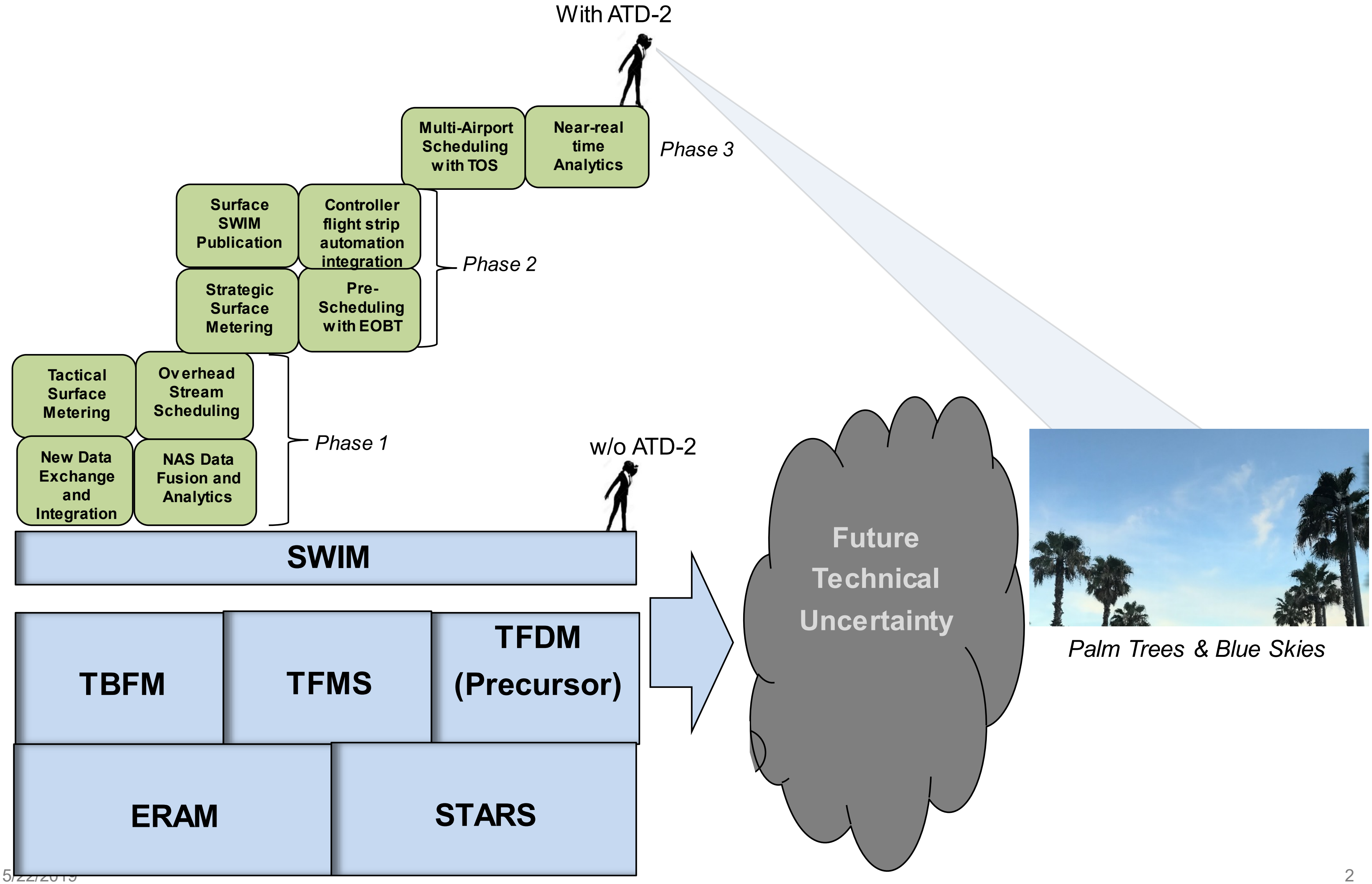
ATD-2 is a Prosumer of SWIM Data

- ATD-2 has greatly benefited from existing SWIM feeds
  - The project is powered by real-time SWIM data

- ATD-2 consumes and utilizes the following SWIM feeds in real-time
  - (TFMS) Traffic Flow Management System - Flight & Flow data
  - (STDDS) SWIM Terminal Data Distribution System
  - (SFDPS) SWIM Flight Data Publication Service
  - (TBFM) Time Based Flow Management
  - (TFDM) Terminal Flight Data Management
  - (TAIS) Terminal Automation Information Service

- ATD-2 produces the following real-time SWIM feed on SWIM R&D
  - TFDM Terminal Publication (TTP)
  - This is in close coordination with the TFDM PO, using same JMSDD
  - The desire is to foster industry ***innovation*** in preparation for TFDM.

ATD-2 Provides a Unique Vantage Point into the Potential Future NAS



With ATD-2

Multi-Airport Scheduling with TOS

Near-real time Analytics

Phase 3

Surface SWIM Publication

Controller flight strip automation integration

Strategic Surface Metering

Pre-Scheduling with EOBT

Phase 2

Tactical Surface Metering

Overhead Stream Scheduling

New Data Exchange and Integration

NAS Data Fusion and Analytics

Phase 1

w/o ATD-2

SWIM

TBFM

TFMS

TFDM (Precursor)

ERAM

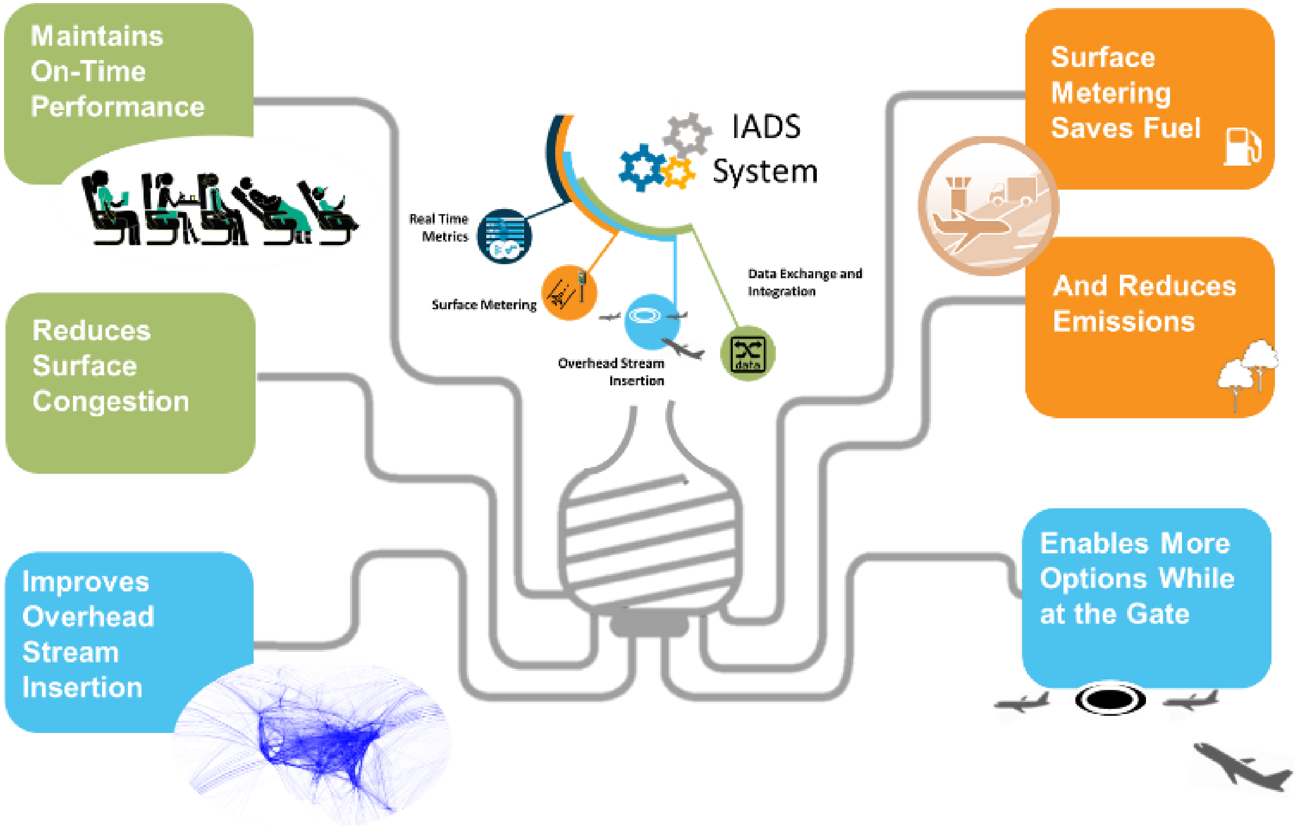
STARS

Future Technical Uncertainty

Palm Trees & Blue Skies

Demonstrating Benefits in the Field

- Multiple benefits mechanisms (benefits through 2019-05-06)
  - 2,295,383 lbs. of fuel saved
  - CO_2 savings equivalent to 82,226 urban trees
  - 270.7 hours of surface delay saved
    - $1,299,413 passenger value of time
    - $368,206 flight crew costs
  - 1,777 hours of reduced runtime on engines



Maintains On-Time Performance

Reduces Surface Congestion

Improves Overhead Stream Insertion

Surface Metering Saves Fuel

And Reduces Emissions

Enables More Options While at the Gate

Achieving Full Benefit from SWIM Flight Data

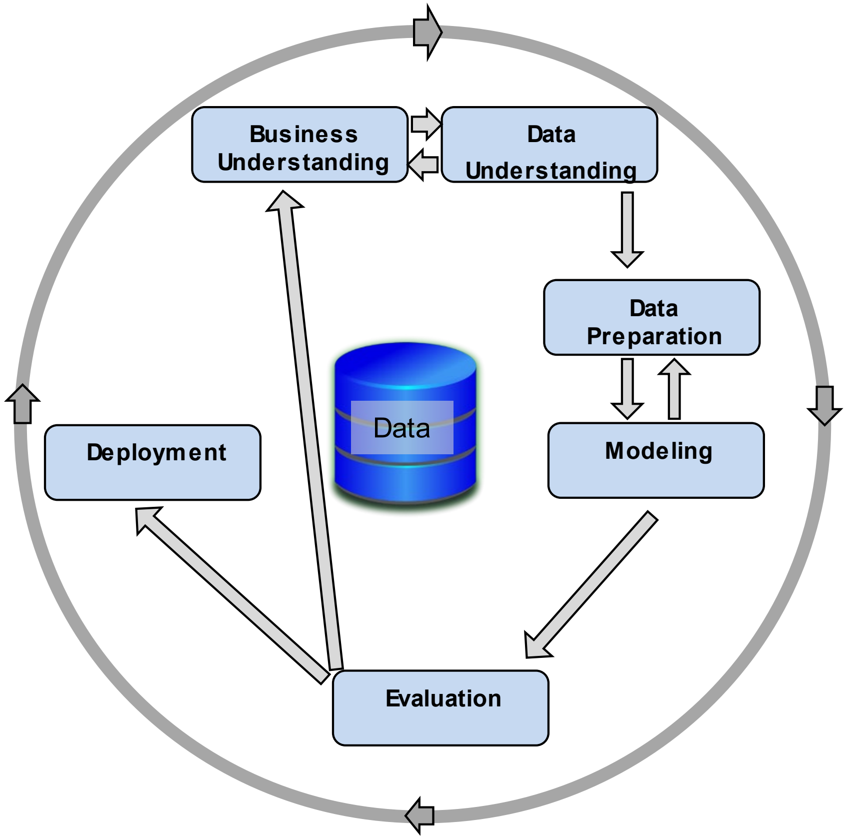
- Many people have worked hard to make SWIM data available (***Thank you***!)
  - Making the data available in a secure, stable platform was a major aviation engineering feat!

- Pre-processing & merging SWIM flight data feeds can be difficult, expensive and error prone
  - FAA decision support systems have valuable output data, but can provide inconsistent information on the same flight that is difficult for consumers to understand
  - Without deep knowledge of the underlying 3T (TFMS, TBFM, TFDM - plus ERAM and STARS) systems, the consumption logic may not lead toward the benefit the community desires
  - If everyone in the aviation industry creates their own SWIM flight data fusion process, many different organizations could come up with different definitions of the 'truth', degrading communication

- The ATD-2 mission required swift progress in field (operational) demonstrations
  - This led to a significant investment in logic that could address SWIM flight data pre-processing and mediation complexities. Much of this work is embodied in the 'Fuser' service.
  - Additional analytical investment was made in post-processing, which evolved over time through an ATD-2 internal data governance process with a feedback loop into the Fuser for more data

- ATD-2 desires to transfer this logic, lessons learned and software (if applicable)
  - After numerous conversations with Industry and FAA, this 'transfer' process is unclear
  - We welcome feedback from you to determine where any additional investments may be warranted
  - The goal is to create the basis for more advanced analytics, which builds upon mediated flight data

Data Mining Requires Agility



- The image above illustrates the Cross-industry standard process for data mining, known as ***CRISP-DM***. This is an open standard process model that describes common approaches used by data mining experts. It is likely the most widely-used analytics model.
- Experts in data mining widely recognize the iterative nature of this process, as well as the need for periodic engagement between business and technical contributors

Airspace Technology Demonstration 2 (ATD-2)

Learning To SWIM with ATD-2

May 22, 2019

Learning to SWIM with ATD-2

- Why ATD-2 Chose to SWIM
- ATD-2 Approach to SWIM
- Overview of Data Feeds
- Data Elements of Interest to ATD-2 in Data Feed
- ATD-2 Lessons Learned

Why ATD-2 Chose to SWIM

- Numerous available products
- Real time
- Single point of access
- More cost effective than legacy alternatives

Why Use Multiple Sources Of Data?

- To cover the entire flight duration and turn-around process
  - If you want the highest quality data available for the entire flight from gate to gate, this requires the use of multiple sources from SWIM
  - In general, the best data comes from the FAA system whose operational mission most closely matches your data need (e.g. if you want strategic constraints and planning info, then TFMS, tactical then TBFM or TFDM)

- Some information only exist, or is higher quality, in certain feeds
  - APREQ Release Times from TBFM
  - EDCT information from TFM Flight Data
  - Surface metering times from TFDM Terminal Publication

- Data redundancy/backup from secondary sources
  - Loss of any one feed still allows data from other feeds to provide value

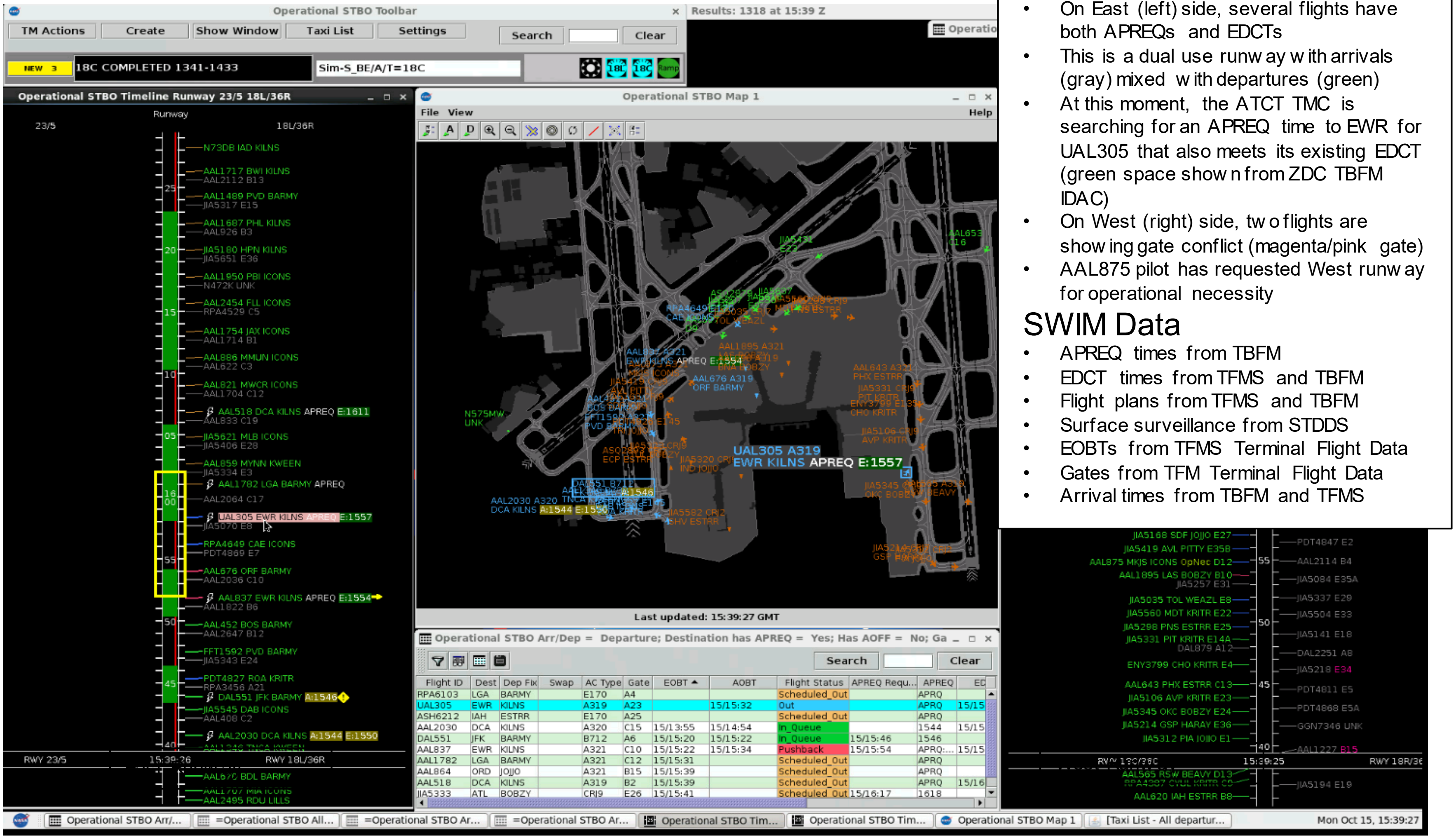
Data Exchange and Integration TMC View

Quick summary (Oct 15, 2018):

- On East (left) side, several flights have both APREQs and EDCTs
- This is a dual use runway with arrivals (gray) mixed with departures (green)
- At this moment, the ATCT TMC is searching for an APREQ time to EWR for UAL305 that also meets its existing EDCT (green space shown from ZDC TBFM IDAC)
- On West (right) side, two flights are showing gate conflict (magenta/pink gate)
- AAL875 pilot has requested West runway for operational necessity

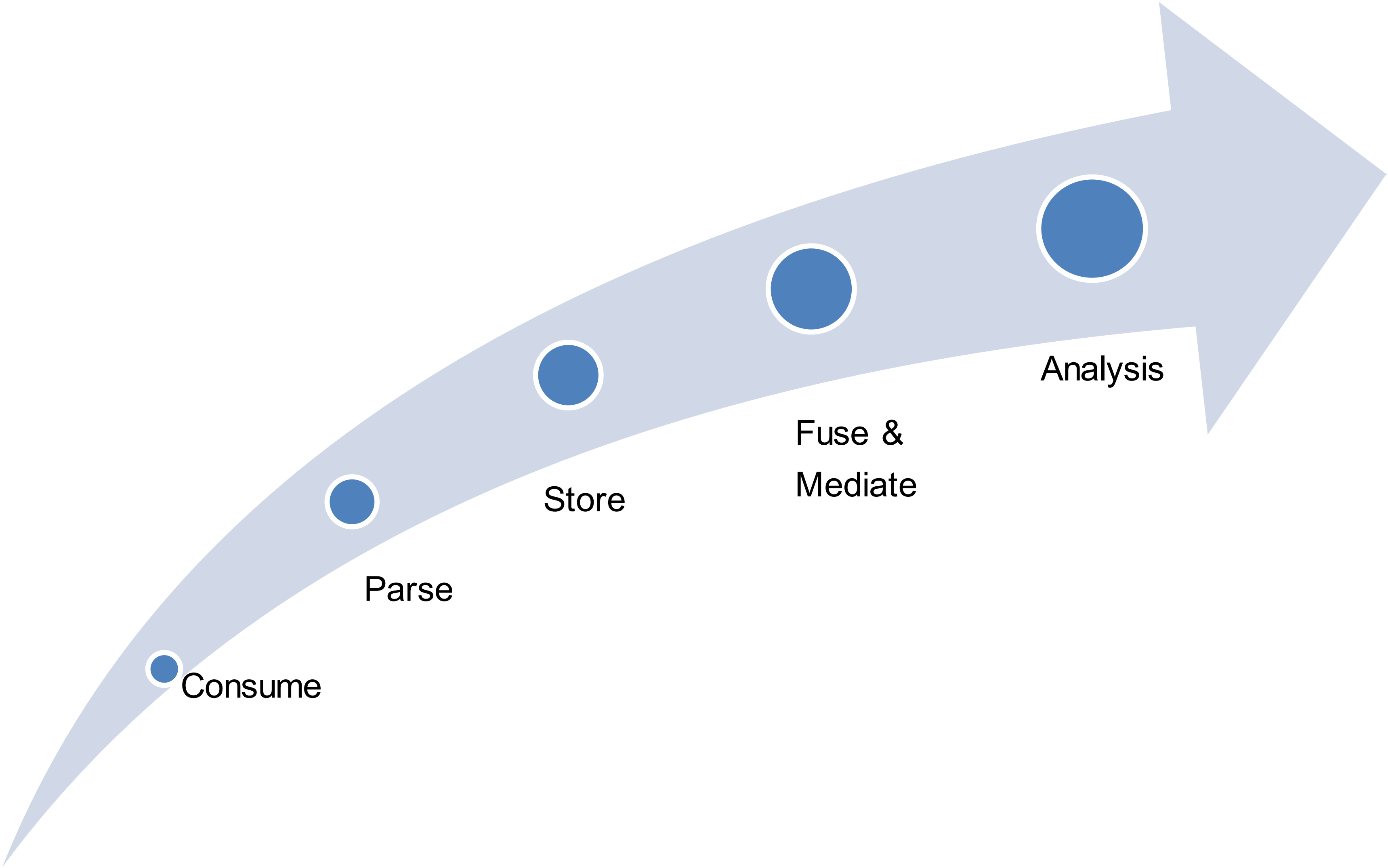
SWIM Data

- APREQ times from TBFM
- EDCT times from TFMS and TBFM
- Flight plans from TFMS and TBFM
- Surface surveillance from STDDS
- EOBTs from TFMS Terminal Flight Data
- Gates from TFM Terminal Flight Data
- Arrival times from TBFM and TFMS



Flight ID	Dest	Dep Fix	Swap	AC Type	Gate	EOBT ▲	AOBT	Flight Status	APREQ Requ...	APREQ	ED
RPA6103	LGA	BARMY		E170	A4			Scheduled_Out		APRQ	
UAL305	EWR	KILNS		A319	A23		15/15:32	Out		APRQ	15/15
ASH6212	IAH	ESTRR		E170	A25			Scheduled_Out		APRQ	
AAL2030	DCA	KILNS		A320	C15	15/13:55	15/14:54	In_Queue		1544	15/15
DAL551	JFK	BARMY		B712	A6	15/15:20	15/15:22	In_Queue	15/15:46	1546	
AAL837	EWR	KILNS		A321	C10	15/15:22	15/15:34	Pushback	15/15:54	APRQ:...	15/15
AAL1782	LGA	BARMY		A321	C12	15/15:31		Scheduled_Out		APRQ	
AAL864	ORD	JOJJO		A321	B15	15/15:39		Scheduled_Out		APRQ	
AAL518	DCA	KILNS		A319	B2	15/15:39		Scheduled_Out		APRQ	15/16
JIA5333	ATL	BOBZY		CRJ9	E26	15/15:41		Scheduled_Out	15/16:17	1618	

Planned Roadmap



Consume

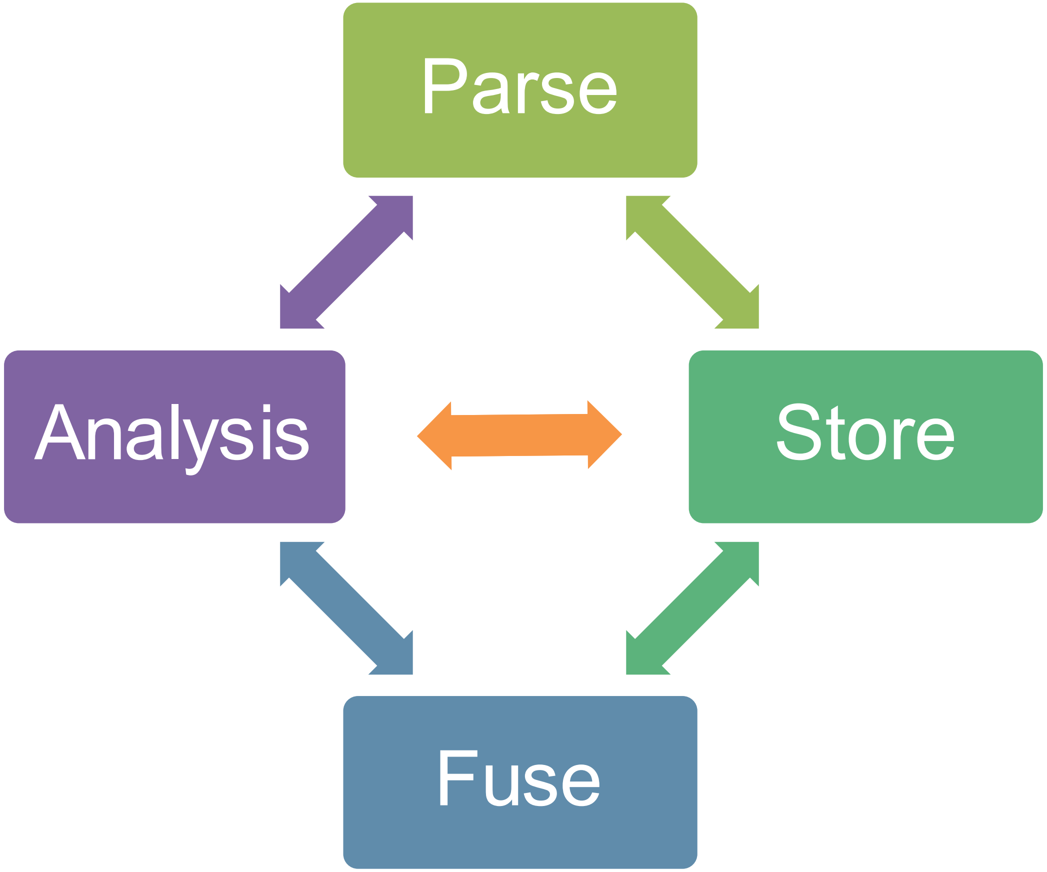
Parse

Store

Fuse & Mediate

Analysis

Actual Roadmap



ATD-2 Steps to SWIM

- Get Familiar with Documentation
- On Ramping
- Consume
- Monitor
- Raw Archives
- Parse
- Create Database
- Match (to be discussed in more detail later in this presentation)
- Fuse (to be discussed in more detail in later presentations)
- Add value (model / scheduler)

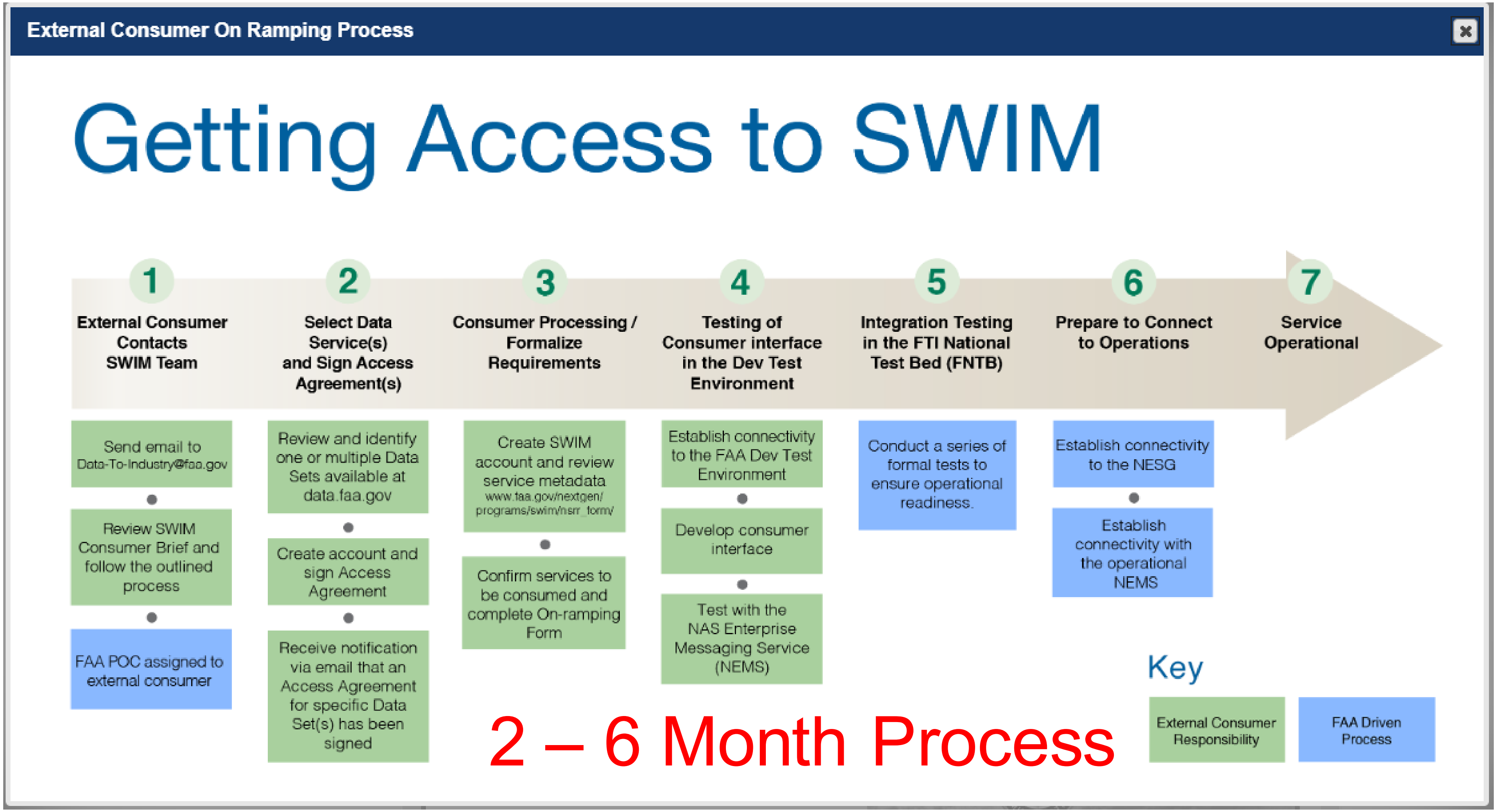
References

- NAS Service Registry and Repository (NSRR)
  - https://nsrr.faa.gov/
- SWIFT Operational Context and Use Cases
  - https://connect.lstechllc.com/index.cfm/main/opconfocusgroup
- FAA NAS Storyboards
  - https://www.faa.gov/about/office_org/headquarters_offices/ang/offices/tc/library/Storyboard/nextgen-overview.html#home
- SWIM Main Page
  - https://www.faa.gov/air_traffic/technology/swim/
- SWIM Users Forum
  - https://www.faa.gov/air_traffic/technology/swim/users_forum/

Accessing SWIM

- Generally Two Options
  - Site to site VPN
    - Up until recently this was your only option
    - Long on ramping process
    - Great when working
    - Can be challenging to troubleshoot with something goes wrong
  - SWIM Cloud Distribution Service (SCDS)
    - Relatively new
    - Very fast on ramping process
    - Does not support request/reply

NEMS via VPN



External Consumer On Ramping Process

Getting Access to SWIM

1 External Consumer Contacts SWIM Team

- Send email to Data-To-Industry@faa.gov
- Review SWIM Consumer Brief and follow the outlined process
- FAA POC assigned to external consumer

2 Select Data Service(s) and Sign Access Agreement(s)

- Review and identify one or multiple Data Sets available at data.faa.gov
- Create account and sign Access Agreement
- Receive notification via email that an Access Agreement for specific Data Set(s) has been signed

3 Consumer Processing / Formalize Requirements

- Create SWIM account and review service metadata www.faa.gov/nextgen/programs/swim/nsrr_form/
- Confirm services to be consumed and complete On-ramping Form

4 Testing of Consumer interface in the Dev Test Environment

- Establish connectivity to the FAA Dev Test Environment
- Develop consumer interface
- Test with the NAS Enterprise Messaging Service (NEMS)

5 Integration Testing in the FTI National Test Bed (FNTB)

- Conduct a series of formal tests to ensure operational readiness.

6 Prepare to Connect to Operations

- Establish connectivity to the NESG
- Establish connectivity with the operational NEMS

7 Service Operational

Key

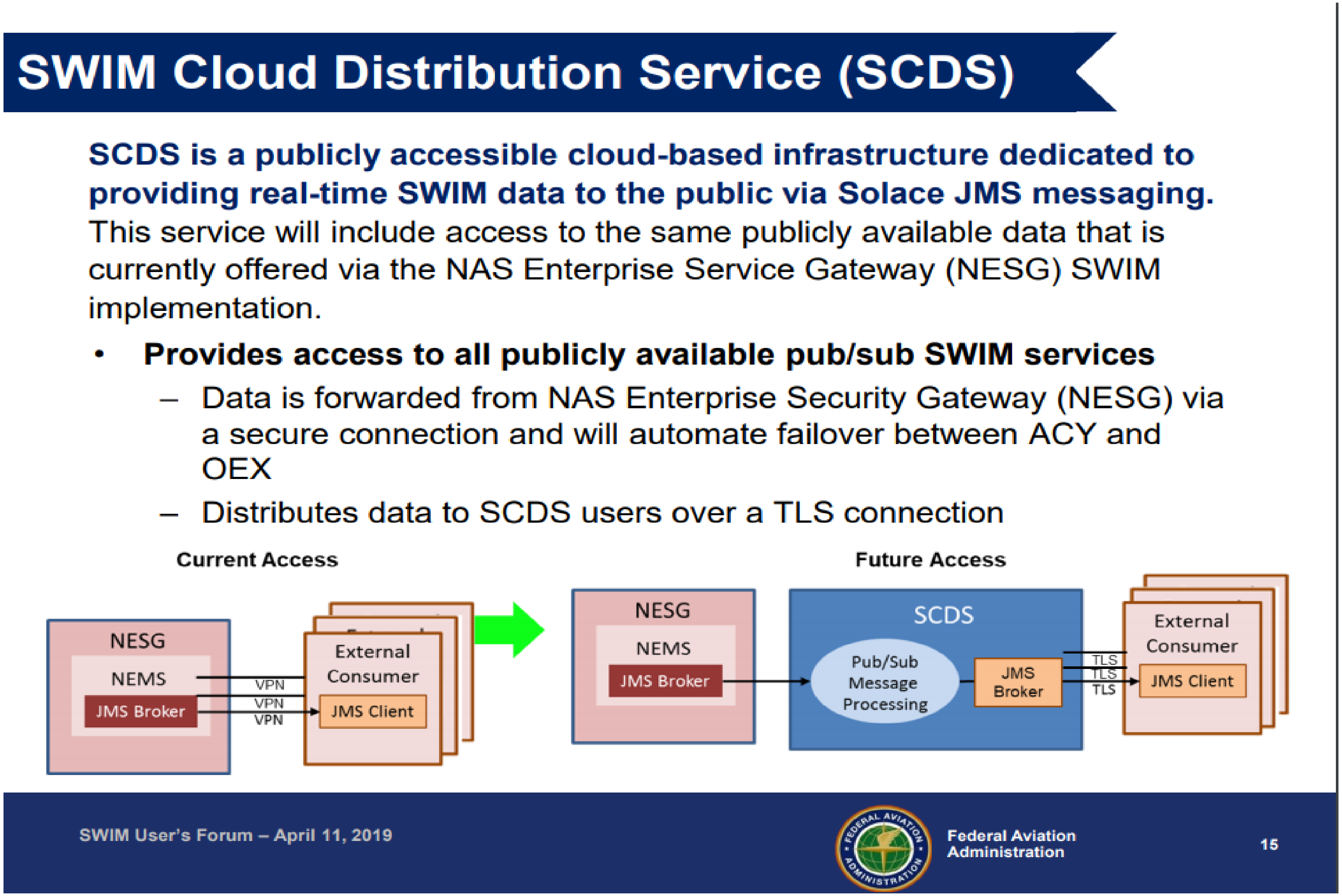
🟩 External Consumer Responsibility

🟦 FAA Driven Process

2 – 6 Month Process

https://www.faa.gov/air_traffic/technology/swim/products/get_connected/#ecbrief

SCDS



SWIM Cloud Distribution Service (SCDS)

SCDS is a publicly accessible cloud-based infrastructure dedicated to providing real-time SWIM data to the public via Solace JMS messaging. This service will include access to the same publicly available data that is currently offered via the NAS Enterprise Service Gateway (NESG) SWIM implementation.

- **Provides access to all publicly available pub/sub SWIM services**
  - Data is forwarded from NAS Enterprise Security Gateway (NESG) via a secure connection and will automate failover between ACY and OEX
  - Distributes data to SCDS users over a TLS connection

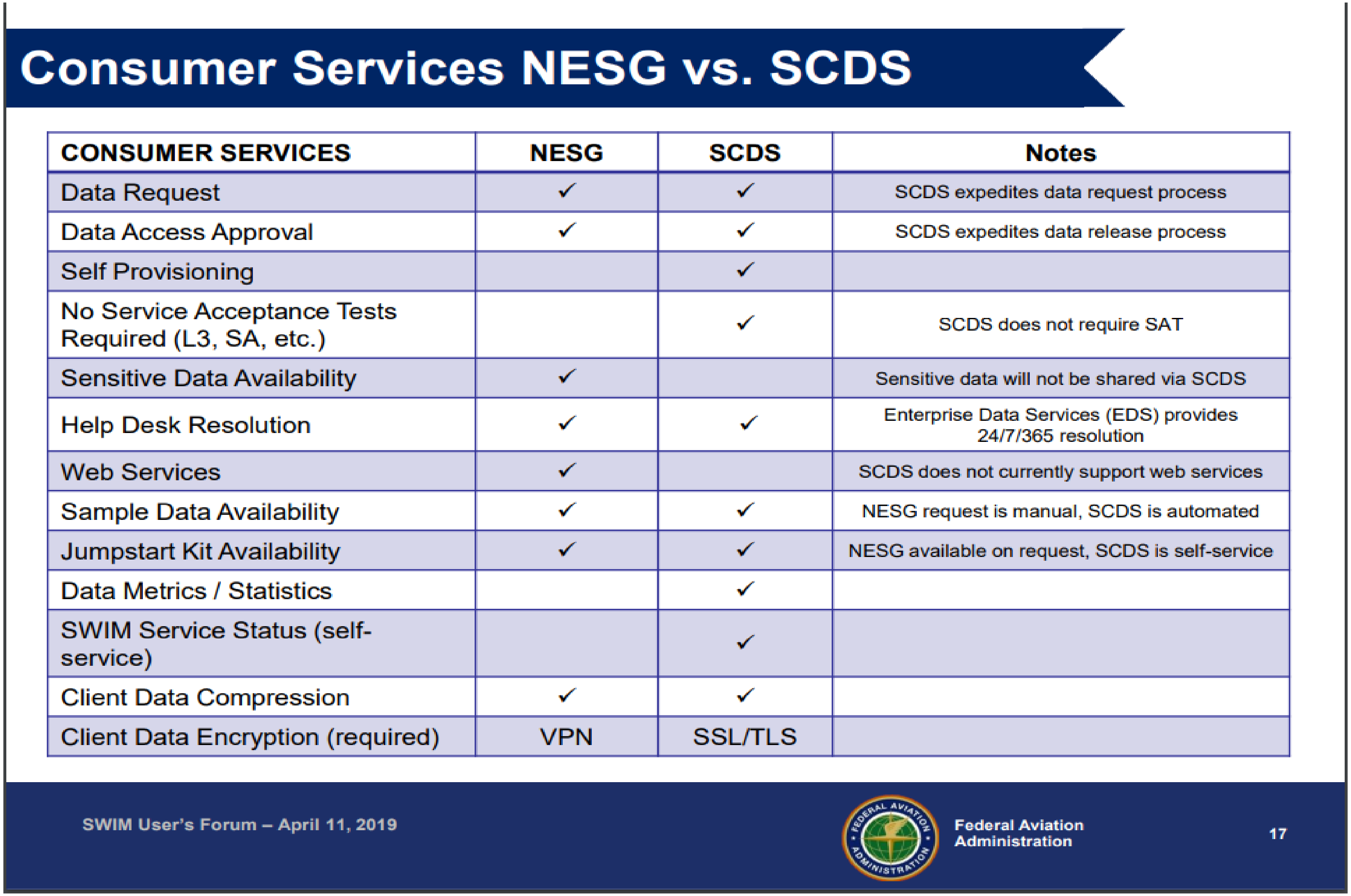
SWIM User's Forum – April 11, 2019

Federal Aviation Administration

15

* Slide copied directly from SWIM Users Forum 24 Briefing

NESG vs SCDS



Consumer Services NESG vs. SCDS

CONSUMER SERVICES	NESG	SCDS	Notes
Data Request	✓	✓	SCDS expedites data request process
Data Access Approval	✓	✓	SCDS expedites data release process
Self Provisioning		✓	
No Service Acceptance Tests Required (L3, SA, etc.)		✓	SCDS does not require SAT
Sensitive Data Availability	✓		Sensitive data will not be shared via SCDS
Help Desk Resolution	✓	✓	Enterprise Data Services (EDS) provides 24/7/365 resolution
Web Services	✓		SCDS does not currently support web services
Sample Data Availability	✓	✓	NESG request is manual, SCDS is automated
Jumpstart Kit Availability	✓	✓	NESG available on request, SCDS is self-service
Data Metrics / Statistics		✓	
SWIM Service Status (self-service)		✓	
Client Data Compression	✓	✓	
Client Data Encryption (required)	VPN	SSL/TLS	

SWIM User's Forum – April 11, 2019

Federal Aviation Administration

17

* Slide copied directly from SWIM Users Forum 24 Briefing

Start Consuming

- Develop your own consumer application
  - Getting started resources
    - NEMS jumpstart kit - https://www.faa.gov/air_traffic/technology/swim/documents/media/user-guide/JumpstartKit-5.1.1.zip
    - SCDS jumpstart kit – available after registering with SCDS

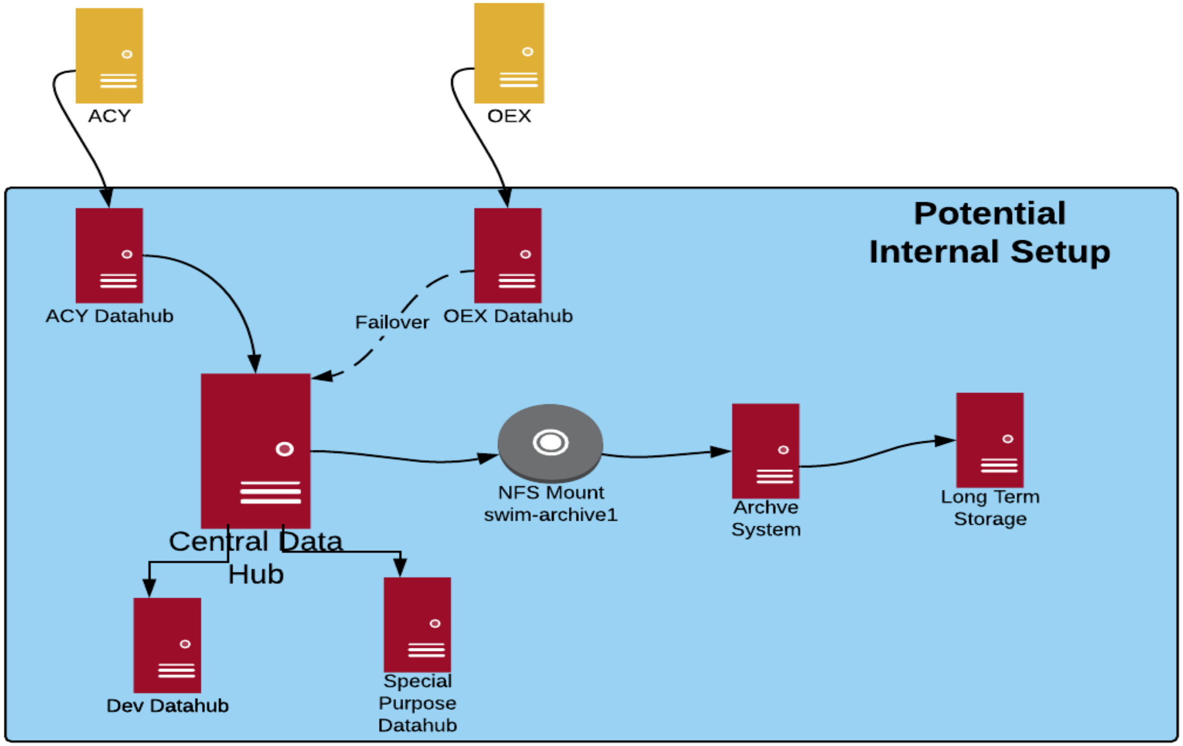
- Use a 3rd party vendor
  - Fastest way to start consuming (if you haven’t developed something already)
  - Bonus, you can skip the testing steps if you are using software from a vendor that has already gone through the process

ATD-2 SWIM Consumer Application

- ATD-2 uses a JMS consumer application
  - Capabilities
    - Connecting to various JMS brokers including
      - Solace
      - Weblogic
      - ActiveMQ
      - IBM MQ
      - AWS SQS
    - Filtering
    - Splitting
    - Recording
    - Repeating
    - Performance Monitoring

Datahub Architecture

- SWIM Data is distributed over queues
  - Can only have one consumer per queue
- Typically an organization will connect and then repeat the data to an internal message bus to allow for other internal functions.



Record the data

- Why
  - Inspect and understand the data
  - Playback
  - Historical archives
  - Recovery option

- Not without effort and cost
  - Need a process for recording the data to files (or other data store)
  - Has to be managed
  - ATD-2 archives TBs of data

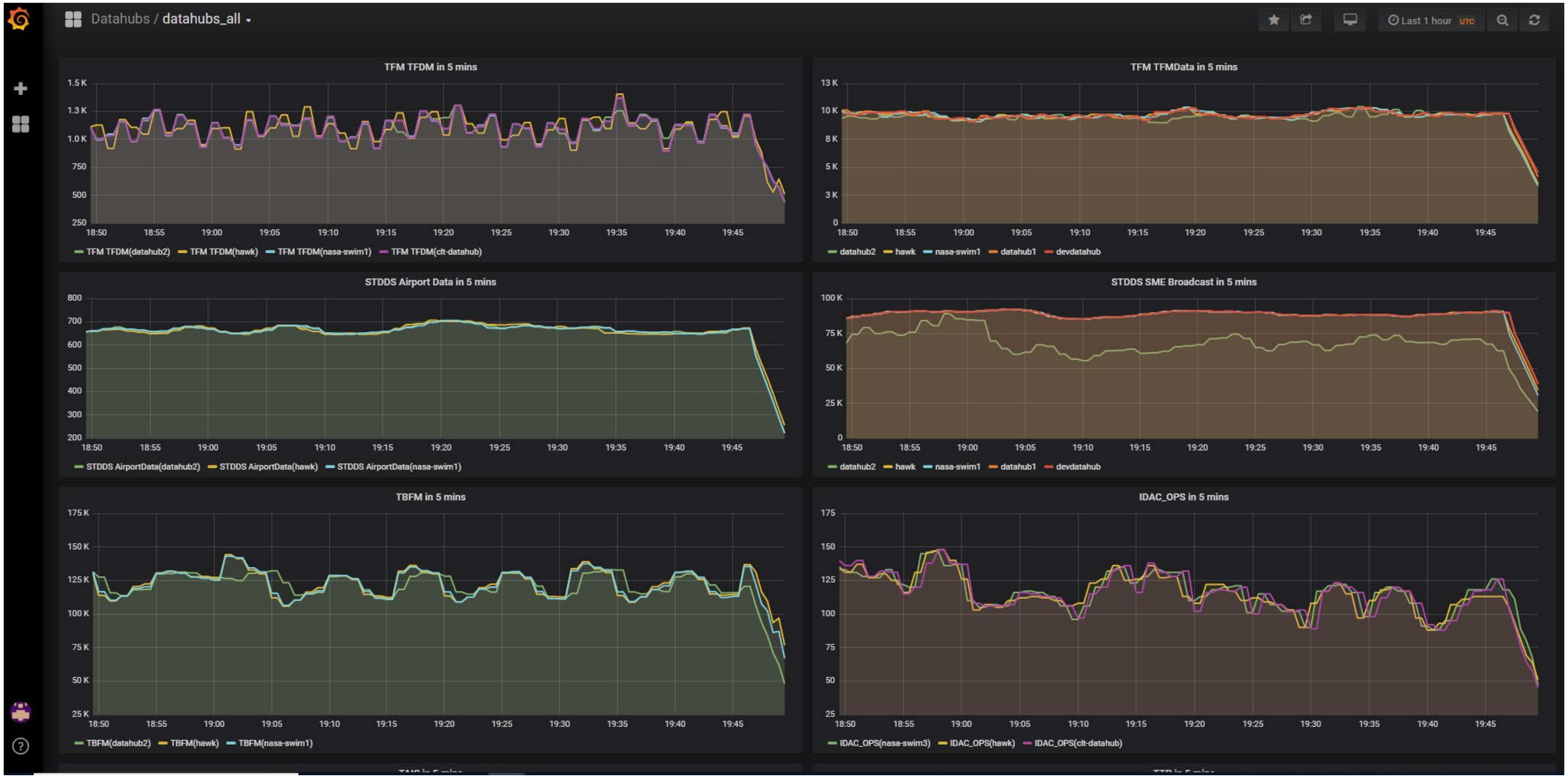
- General guidance
  - Capture headers and message content
  - Store by hour
  - Organize by year, month, day
  - Compress the files
  - Prepare a lot of storage space or have an expandable storage solution
    - Never met an analyst willing to get rid of data

Monitoring

- Grafana Reports
- System Monitor
- Prometheus Alerts/Emails

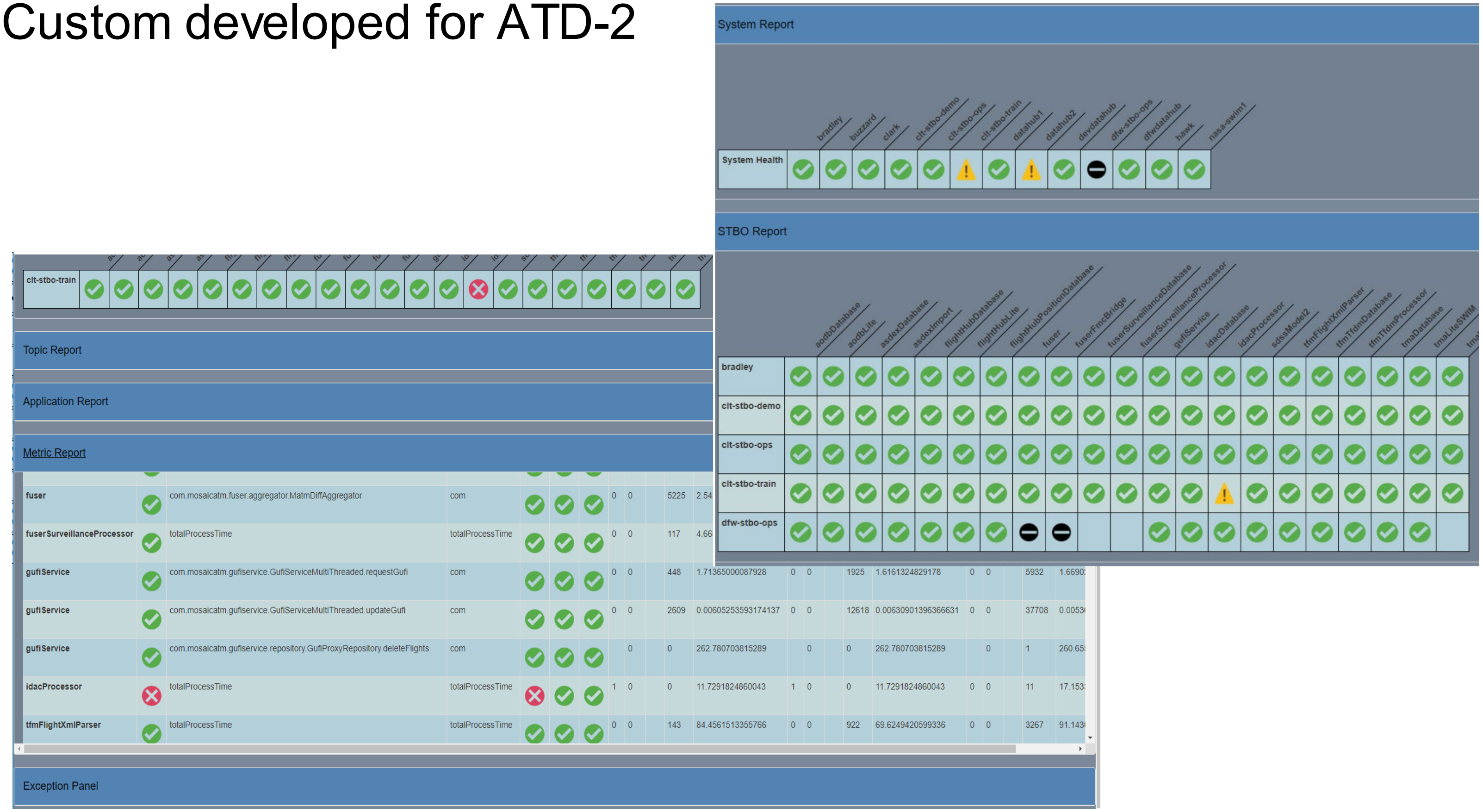
Grafana

- Monitors feeds over time
- Allows us to compare feeds from multiple sources (ACY and OEX)



System Monitor

- Quick look view of the status of key ATD-2 Systems
- Ability to drill down to the specific metrics
- Custom developed for ATD-2



System Report

	bradley	buzzard	clark	clt-stbo-demo	clt-stbo-ops	clt-stbo-train	datahub1	datahub2	devdatahub	dfw-stbo-ops	dfwdatahub	hawk	nasa-swim1
System Health	✓	✓	✓	✓	✓	⚠	✓	⚠	✓	⛔	✓	✓	✓

STBO Report

	aodbDatabase	aodbLite	asdexDatabase	asdexImport	flightHubDatabase	flightHubLite	flightHubPositionDatabase	fuser	fuserFmcBridge	fuserSurveillanceDatabase	fuserSurveillanceProcessor	gufiService	idacDatabase	idacProcessor	sdssModel2	tfmFlightXmlParser	tfmTfdmDatabase	tfmTfdmProcessor	tmaDatabase	tmaLiteSWIM	tma
bradley	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
clt-stbo-demo	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
clt-stbo-ops	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
clt-stbo-train	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	⚠	✓	✓	✓	✓	✓	✓	✓
dfw-stbo-ops	✓	✓	✓	✓	✓	✓	✓	⛔	⛔			✓	✓	✓	✓	✓	✓	✓	✓	✓	

	ac	ac	as	as	fli	fli	fli	fu	fu	fu	fu	gu	id	id	sc	tfi	tfi	tfi	tn	tn	tn
clt-stbo-train	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	☒	✓	✓	✓	✓	✓	✓	✓

Topic Report

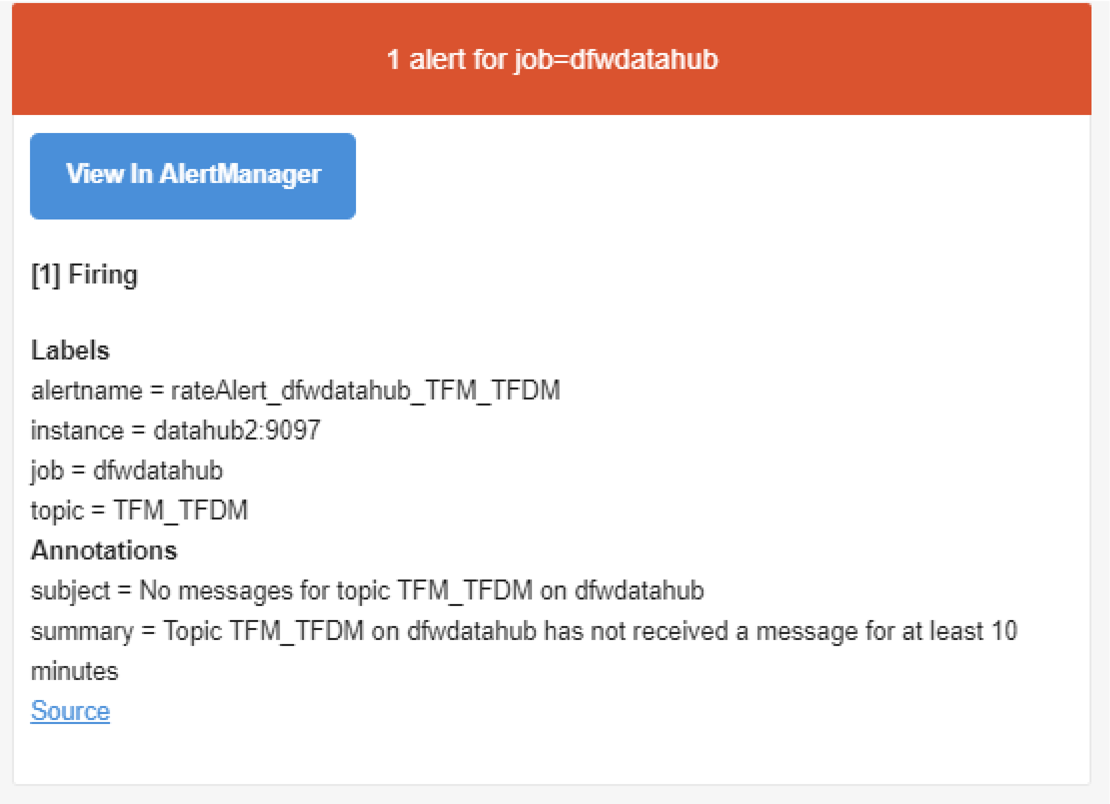
Application Report

Metric Report

fuser	✓	com.mosaicatm.fuser.aggregator.MatmDiffAggregator	com	✓	✓	✓	0	0		5225	2.54										
fuserSurveillanceProcessor	✓	totalProcessTime	totalProcessTime	✓	✓	✓	0	0		117	4.66										
gufiService	✓	com.mosaicatm.gufiservice.GufiServiceMultiThreaded.requestGufi	com	✓	✓	✓	0	0		448	1.71365000087928	0	0		1925	1.6161324829178	0	0		5932	1.6690
gufiService	✓	com.mosaicatm.gufiservice.GufiServiceMultiThreaded.updateGufi	com	✓	✓	✓	0	0		2609	0.00605253593174137	0	0		12618	0.00630901396366631	0	0		37708	0.0053
gufiService	✓	com.mosaicatm.gufiservice.repository.GufiProxyRepository.deleteFlights	com	✓	✓	✓		0		0	262.780703815289		0		0	262.780703815289		0		1	260.65
idacProcessor	☒	totalProcessTime	totalProcessTime	☒	✓	✓	1	0		0	11.7291824860043	1	0		0	11.7291824860043	0	0		11	17.153
tfmFlightXmlParser	✓	totalProcessTime	totalProcessTime	✓	✓	✓	0	0		143	84.4561513355766	0	0		922	69.6249420599336	0	0		3267	91.143

Exception Panel

Prometheus Monitoring Emails



[1] Firing

Labels

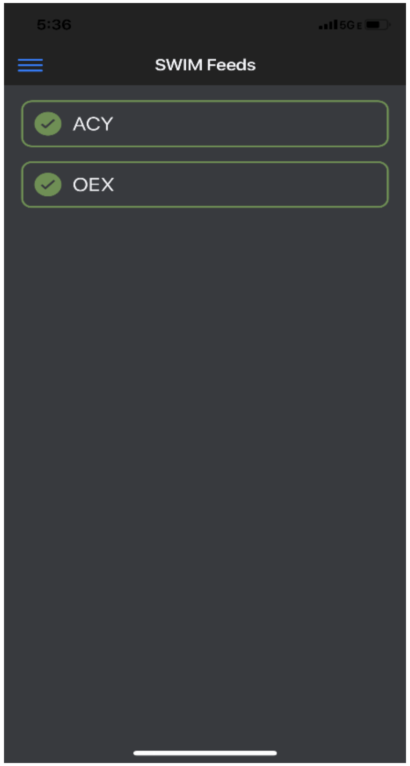
alertname = rateAlert_dfwdatahub_TFM_TFDM
instance = datahub2:9097
job = dfwdatahub
topic = TFM_TFDM

Annotations

subject = No messages for topic TFM_TFDM on dfwdatahub
summary = Topic TFM_TFDM on dfwdatahub has not received a message for at least 10 minutes
Source

Mobile App

- Swim Feed Dashboard
  - IOS - https://itunes.apple.com/us/app/swim-feed-dashboard/id1453740223
  - Android https://play.google.com/store/apps/details?id=us.mosaicsoftware.swimfeeddashboard&hl=en_AU
  - Reports status of Mosaic SWIM Feeds
    - Could be helpful in determining if your SWIM connection is the problem or more upstream impacting everyone

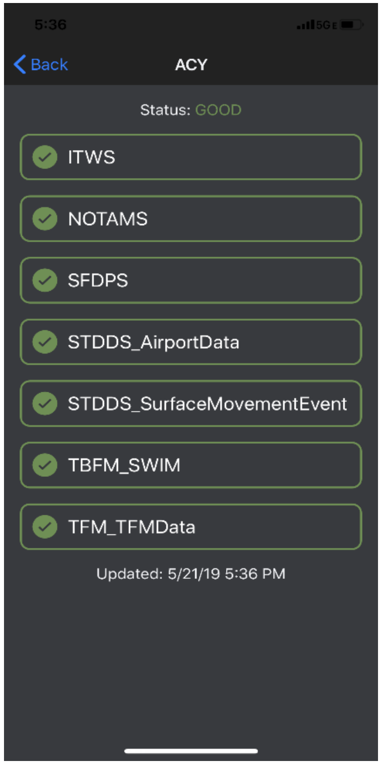


5:36

SWIM Feeds

✓ ACY

✓ OEX



5:36

Back

ACY

Status: GOOD

✓ ITWS

✓ NOTAMS

✓ SFDPS

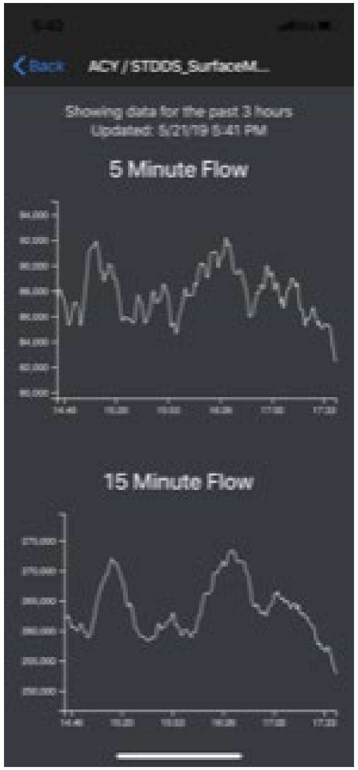
✓ STDDS_AirportData

✓ STDDS_SurfaceMovementEvent

✓ TBFM_SWIM

✓ TFM_TFMData

Updated: 5/21/19 5:36 PM



Back

ACY / STDDS_SurfaceM...

Showing data for the past 3 hours

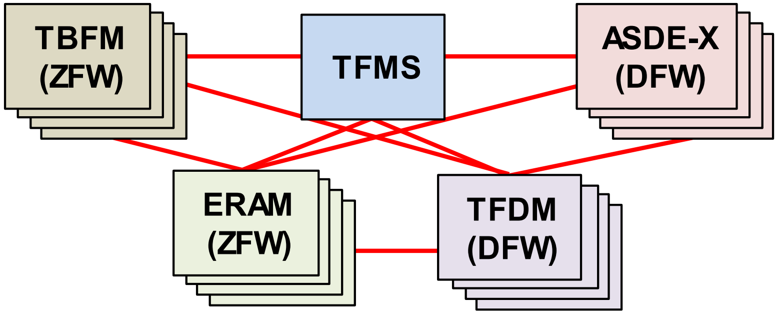
Updated: 5/21/19 5:41 PM

Overview of Data Feeds

Feed	Data Source	ATD-2 Use	Data
TfmData Flight	TFMS	• Schedule data • CDM data • Flight plans • EDCTs • Track data	Flight
STDDS SMES	ASDE-X/ASSC	• Surface track data	Track
TBFM MIS	TBFM	• Release times • EDCTs	Flight Plan
TfmData Terminal	TFMS	• EOBTs • Gates • Tail Numbers	Flight
TfmData Flow	TFMS	• Ground Stops • MIT restrictions (planned)	TMIs
SFDPS	ERAM	• Support for NASA research TBFM	Flight Plan

Overview of Data Feeds

- The NAS systems generating flight data SWIM feeds are interconnected.
  - Basic understanding the NAS architecture helps make sense of the SWIM data.



NAS System	SWIM Feeds	Primary Inputs	Secondary Inputs	Deployments
TFMS	• TfmData Flight • TfmData Flow • TfmData Terminal	• ERAM flight/track data • OAG schedule • Airline CDM messages • ATOP oceanic track	• TBFM release times • STDDS surface times • OIS / NTML • International feeds • TFDM predictions	1
ERAM	• SFDPS	• ATC flight plan data • En route radar track	• TFMS EDCTs • TFMS reroutes	1 per ARTCC (20 total)
TBFM	• TBFM MIS	• ERAM flight/track data (adjacent ARTCCs) • TRACON fight/track data	• TFMS international track • TFDM release time negotiation	1 per ARTCC (20 total)
ASDE-X	• STDDS SMES	• Surface radar track • ERAM flight data		1 per airport (38 total)
TFDM	• TTP	• All the above		1 per airport (future)

Overview of Data Feeds

Feed	Daily Volume	Sync Messages?	Diff or Full Messages
TfmData Flight	280k (note: much larger batches)	No	Full
STDDS SMES	22.5 million	No	Diff (with full at regular intervals per flight)
TBFM MIS	25 million	Yes	Diff
TfmData Terminal	225k	No	Full
TfmData Flow	1.7 million	Yes	Full
SFDPS	5.3 million	No	Full

Overview of Data Feeds

Source	Flights Included	Earliest Flight Data	Latest Flight Data
TFMData Flight	All IFR	~24 hours before scheduled OAG operation	"Actual" arrival gate time published by Operators (up to 2 hours after taxi in)
TFMData Terminal	Only flights that publish CDM messages including Terminal Flight Data elements	~24 hours before scheduled operation	"Actual" arrival gate time published by Operators (up to 2 hours after taxi in)
SFDPS	All IFR and some VFR	Filed flight plan	Flight lands
TBFM MIS	Impacted by a TBFM arrival (TMA) or departure (EDC) system	Flight plan filed within TBFM adapted region	Track ends or leaves TBFM adapted region
STDDS SMES	Operating at or near ASDEX airport	First correlated flight track	Last correlated flight track

Overview of Data Feeds
SWIM Feed IDs

Source	Unique Flight IDs	Examples	ID Recycles	Multiple IDs per Flight
TFMData Flight	flightRef	100725389	Infrequent	Rare
	gufi	KT44707500	Infrequent	Yes
TBFM MIS	tmaId	T06629	Immediately	Yes
SFDPS	fdpsGufi	us.fdps.2019-05-09T13:40:40Z.000/19/501	Never	Rare
	uuidGufi	b443e49c-0cdf-47ed-bce5-5275a54a8cc0	Unlikely	Yes, on failover
	flightPlanIdentifier	KT44707500	Infrequent	Yes
STDDS SMES	stid	1716539	Infrequent	Possible
	track	862	Frequent	Frequent
	enhancedData.eramGufi	KT44707500	Infrequent	Yes
	enhancedData.sfdpsGufi	us.fdps.2019-05-09T13:40:40Z.000/19/501	Never	Rare

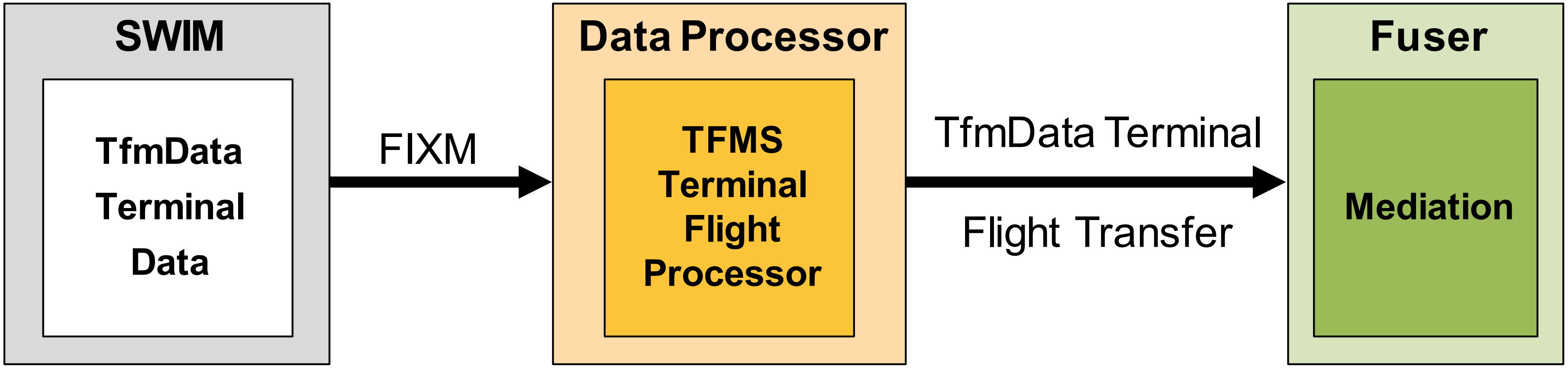
Data Elements in SWIM Feeds

- Full mappings available publically here:
  - https://aviationsystems.arc.nasa.gov/atd2-industry-days/fuser/TFMS-Flight-Data-Mapping_85328230.html

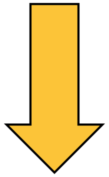
- Mapping Example – TfmData Flight Messages

Data Element	TfmData Flight Message Type	TfmData Flight Data Element
acid	<all>	qualifiedAircraftId.aircraftId
aircraftType	BOUNDARY_CROSSING_UPDATE DEPARTURE_INFORMATION FLIGHT_CREATE FLIGHT_MODIFY FLIGHT_PLAN_AMENDMENT_INFO FLIGHT_PLAN_INFORMATION FLIGHT_ROUTE FLIGHT_SCHEDULE_ACTIVATE FLIGHT_TIMES	flightAircraftSpecs flightStatusAndSpec.aircraftspecification airlineData.flightStatusAndSpec.aircraftspecification airlineData.flightStatusAndSpec.aircraftspecification newFlightAircraftSpecs flightAircraftSpecs flightStatusAndSpec.aircraftspecification flightStatusAndSpec.aircraftspecification flightStatusAndSpec.aircraftspecification

Parsing Data Feeds



flight.getDepartureNas().getStandPositionAndTime().getStandTime().
getActual().getTime()



flight.getActualOffBlockTime()

Parsing Data Feeds

- One parser per feed
  - Transforms raw XML file into flattened object structure
    - Leveraging JAXB with jaxb2-basics to simply XML->Java Object conversion
  - In some cases, perform aggregation against messages from the same source

- Aggregation examples
  - STDDS SMES
    - Track data is sent as diffs
    - Have to aggregate with previous messages for same flight so that complete position update is processed by Fuser

	Previous	Current	To Fuser
Time	12:00:01	12:00:02	12:00:02
Latitude	35.2156	-----	35.2156
Longitude	-80.947**3**	-80.947**5**	-80.947**5**

  - TBFM-MIS
    - Need to be able to handle SYNC messages properly

- Parsers archive raw message to databases for analysis

TFMData Flight Parser

- Parser features
  - Message splitting/filtering
    - Filter to MessageType=FlightDataMessageOutgoing.
    - Un-batch incoming messages.
  - Message aggregation
    - Sort messages within each by time order.
    - Messages are full, but message types contain different types of data.
  - Message matching
    - Global GUFI assigned from ATD2 GufiService.
    - Locally match data on Flight Ref with additional internal validation.
  - Message Transformation
    - Common format with GUFI and all possible data across TFM message types.

- Key headers
  - MessageType : FlowInformationMessageOutgoing / FlightDataMessageOutgoing

TFMData Flight Parser

- Lessons learned
  - TFMS Flight Ref is good for matching NAS-wide flight data.
  - Schema is somewhat challenging to parse -- most messages are different format.
  - Message types (and message trigger) are important to interpreting the message intent.
    - E.g. MsgType=FLIGHT_TIMES, Trigger=NEMS_TBFM_FLT_DEPARTURE_MSG
  - Some JMSDD data elements are not available yet (e.g. etdTimeType=METERED).

STDDS SME Parser

- Parser features
  - Message splitting/filtering
    - Process each airport independently.
  - Message aggregation
    - Merge input data on track number.
    - Handle "full=true" sync messages.
    - Handle the delete element (r=1) attribute.
  - Message matching
    - Global GUFI assigned from ATD2 GufiService.
    - Locally match data on STDDS Surface Track ID with additional internal validation.
  - Message Transformation
    - Common format with value-added fields (e.g. GUFI).

- Key headers
  - airport
  - msgType : AT (PositionReport), AY (SystemStatus), AD (adsbReport), ML (mlatReport)

STDDS SME Parser

- Lessons learned
  - Beacon code used to retrieve core data from ERAM (e.g. ACID, aircraft type, etc).
    - When aircraft emit the “wrong” beacon code, ACID will also be wrong.
  - Track messages are interpolated (may indicate wrong location).
  - Schema updates are deployed to different airports at different times.

TBFM-MIS Parser

- Parser features
  - Message filtering
    - ATD2 filters to messages from TBFM ARTCCs of interest.
  - Message aggregation
    - Merge input data on TMA ID.
    - Merge elements in sta, eta, sch, mrp data groups by common “mfx” name.
    - Handle NEW, AMD, DEL message types.
    - Handle messages out of order.
    - Handle sync messages.
  - Message matching
    - Global GUFI assigned from ATD2 GufiService.
    - Locally match data on TMA ID with additional internal validation.
  - Message Transformation
    - Common format with value-added fields (e.g. GUFI).

- Key headers
  - ARTCC – TBFM originating ARTCC.
  - SYNC – sync message.
  - STDCHG – indicator that release time either set or unset.

TBFM-MIS Parser

- Lessons learned
  - TmaId used for merging data, but can recycle very quickly.
  - TBFM SYNC messages had to be handled as syncs and not updates.
    - Sync messages can take many minutes to complete. Prevented resetting the EDCT back to an old value.
  - Treat AMD/NEW the same.

Additional References

- Animated storyboards
  - About ERAM in general
  - About TBFM in general
  - About TFMS in general
  - About TFDM in general

- Operation Context and Use Cases
  - About TfmData Flow Operational Context and Use Cases
  - About TfmData Flight Operational Context and Use Cases
  - About TBFM SWIM Operational Context and Use Cases
  - About SFDPS Operational Context and Use Cases

Wrap Up

- SWIM contains lots of data
- The data has lots of value
- But consuming that data is challenging
- If only there were some way to fuse the data together…

Questions

Airspace Technology Demonstration 2 (ATD-2)

Fuser Why Everyone Should Have One

May 22, 2019

Why Have a Fuser

- One stop shopping for all your flight data needs

By HikingArtist.com

Fuser Helps Solves SWIM Challenges

- Problem
  - With the great new FAA SWIM feeds, your organization feels they are drowning in data that they do not understand.
  - You need to make ***actionable, operational information*** out of this data!

- Solution
  - To accomplish this, you need a framework that can mediate between disparate sources of data, pulling in the ***right data, at the right time***.
  - Need information on which data source is best to use for a specific need.
  - Access to the information in common well defined data model

5/22/2019

Design Motivations

- Reduce time spent troubleshooting
  - Less time troubleshooting = more time to create new capabilities
- Needed a system where the data is exposed at all critical phases
  - What we received
  - What we parsed
  - How we matched/correlated
  - How we fused
- More data is not a linear problem. As you get more data feeds the possible issues become exponential
- Tired of seeing same issues manifest on various efforts and phases
  - Redundant effort
  - Inconsistent results
  - Inconsistent definitions

5/22/2019

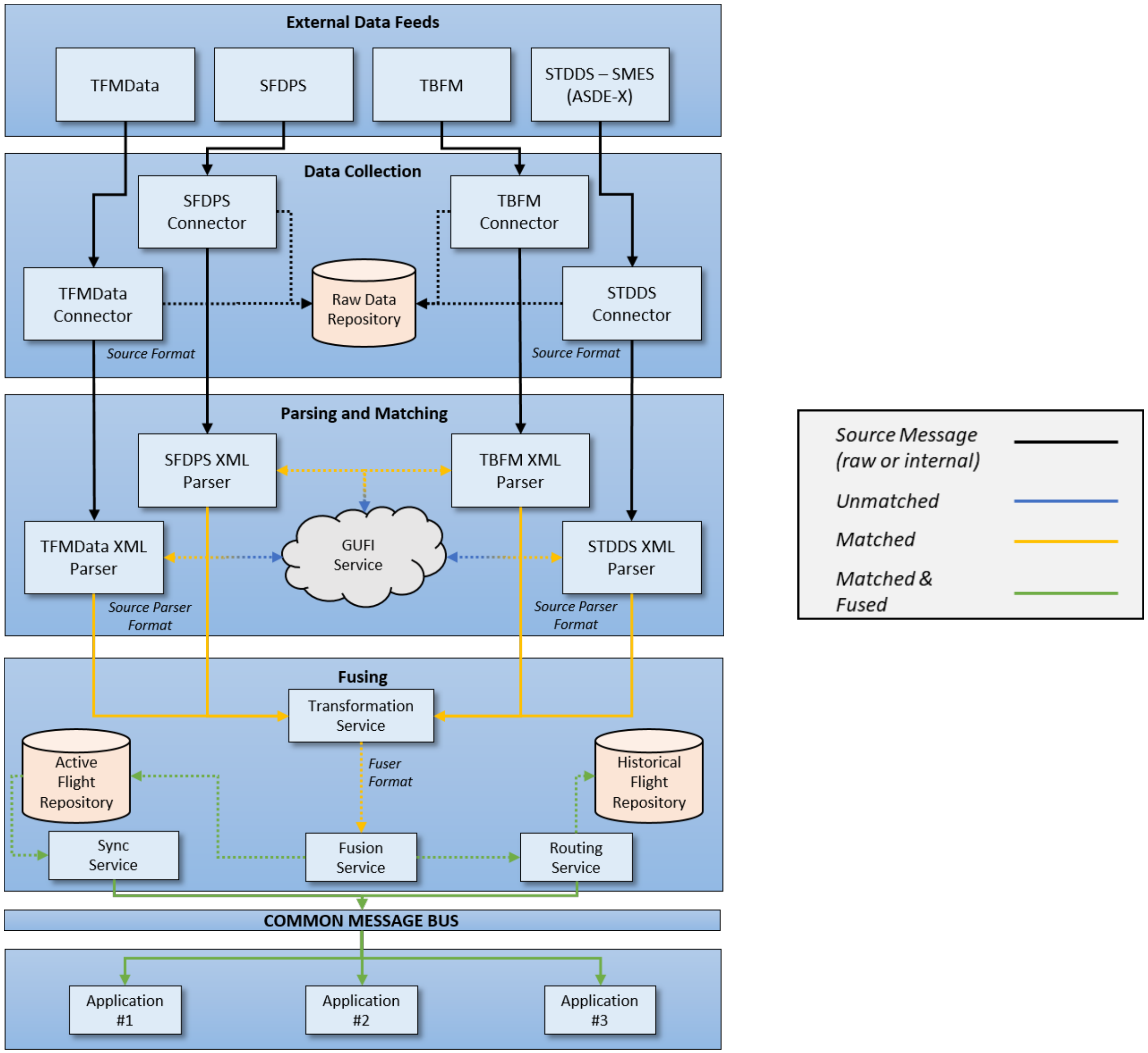
What is the Fuser

- Fuser is a system composed of multiple components providing
  - Parsers for various data sources
  - Matching Services
  - Fusion Services
    - Transformation
    - Filtering
    - Merging
    - Mediation
  - Database Loaders

- Common well defined schema

5/22/2019

Fuser High Level Architecture



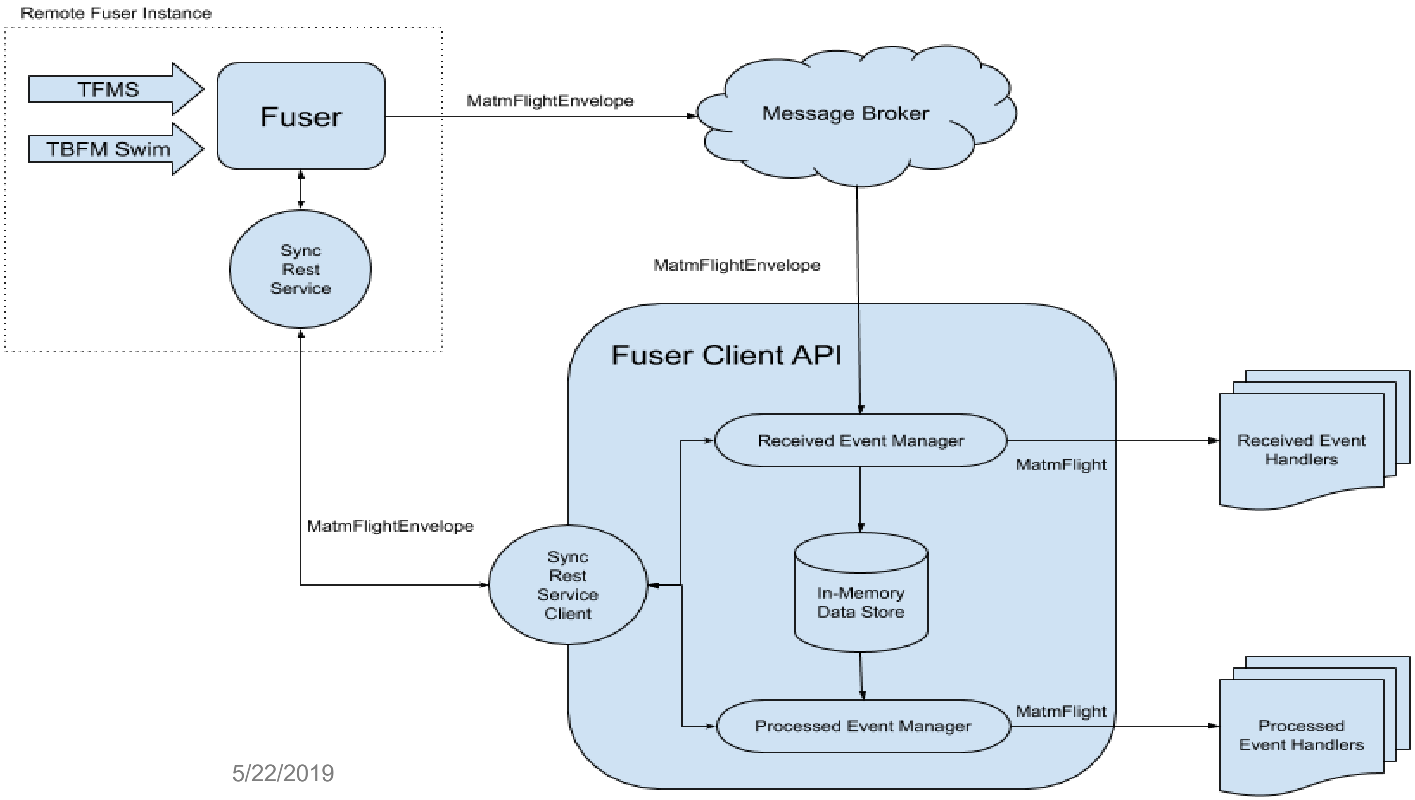
Additional Capabilities

- Flight Management
  - Create
  - Update
  - Remove
- Data Distribution
- Built in performance monitoring
- Built in recovery solution
- Reconstitution available for Fuser clients
- Pluggable
  - Pluggable architecture for new data feeds

5/22/2019

Fuser Client API

- Quickly develop Fuser driven applications
- Handles connecting and managing the connection
- Creates an in memory repository
- Support for event listeners



Technology Stack

- Java
- Apache Camel
- Spring Framework
- Apache CXF
  - Web services used for reconstitution
- Redis
  - Used for recovery
- ActiveMQ
  - Pub/Sub messaging
- Postgresql
  - Optional database loading
- Dropwizard
  - Metrics

5/22/2019

Fuser Industry Documentation

- Fuser Overview
- Component Descriptions
- Fuser Data Dictionary
  - Complete list of fields found in the Fuser
  - Field description
    - Most fields were meant to be self documenting
- SWIM data to Fuser Mappings
  - Likely more valuable than the data dictionary itself
- References to other SWIM related documentation
- https://aviationsystems.arc.nasa.gov/atd2-industry-days/fuser/ATD-2-Industry-Day-Documentation-Outline_81565170.html

5/22/2019

Questions

Airspace Technology Demonstration 2 (ATD-2)

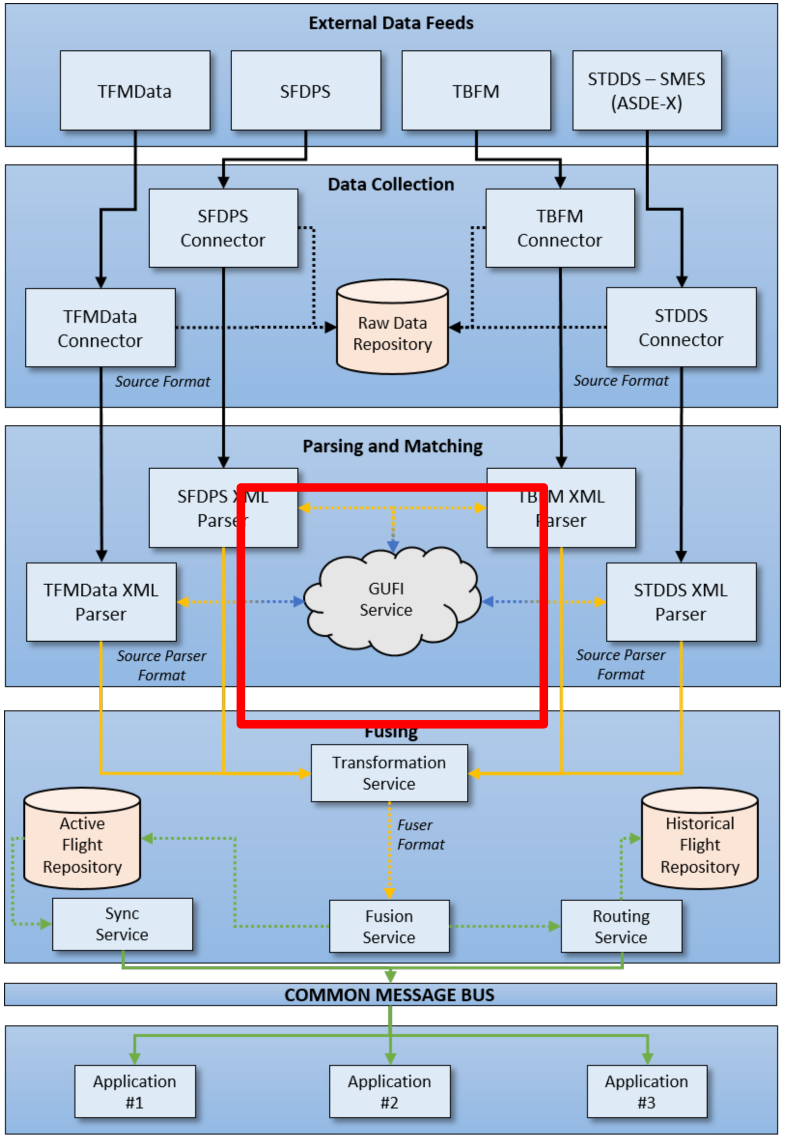
Fuser Deeper Dive (Mediation & Use Cases)

May 22, 2019

Discussion Topics

- Flight Matching
- Fuser Processing
  - Transformation
  - Filtering
  - Mediation
    - Fuser Metadata
    - Use Cases

Flight Matching



Flight Matching: Overview

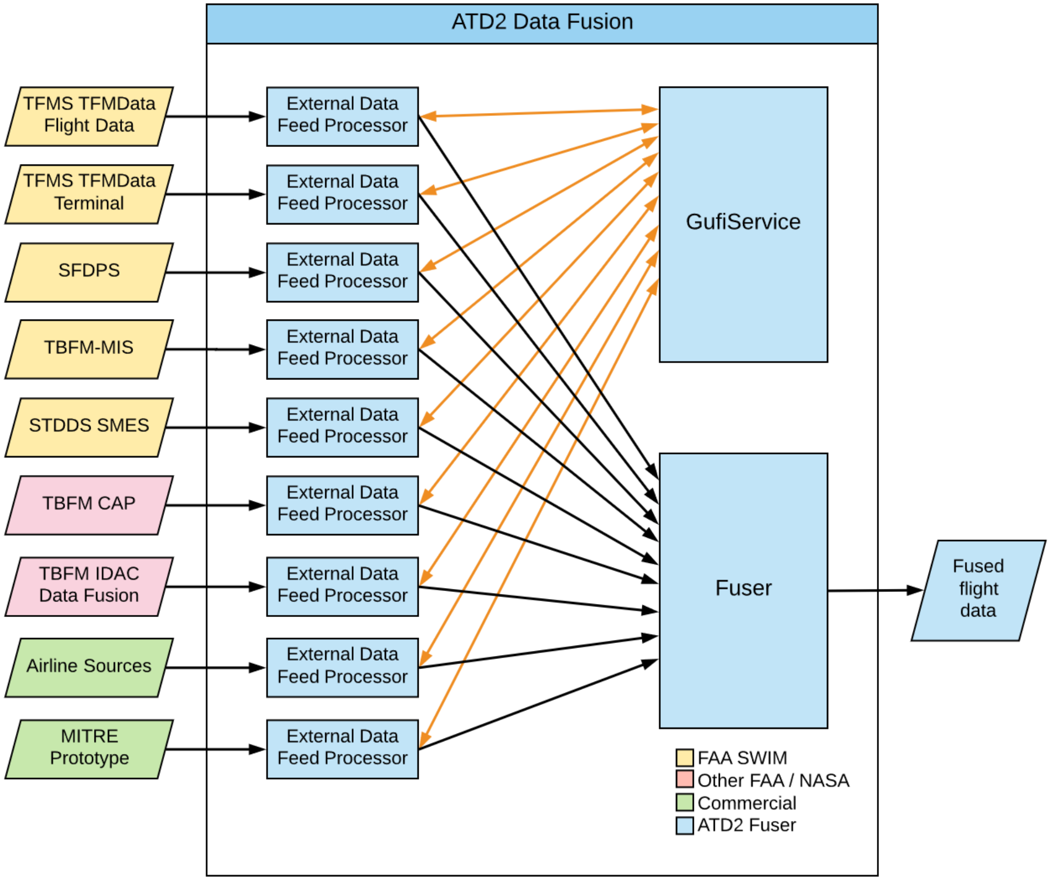
- Goal
  - Assign a Global Unique Flight Identifier (GUFI) to every flight message.
    - Ex: AAL1428.DFW.MCI.190507.1504.0132.TFM

- Why?
  - GUFI links together flight data across all external data feed sources.
  - Crucial precursor to data fusion.

- What is a flight?
  - Flight matching is where the “flight” is defined.
  - ATD2 defines a “flight” as the full lifecycle : starting with the scheduled or planned operation and ending at the arrival gate.

Flight Matching: ATD2 Approach

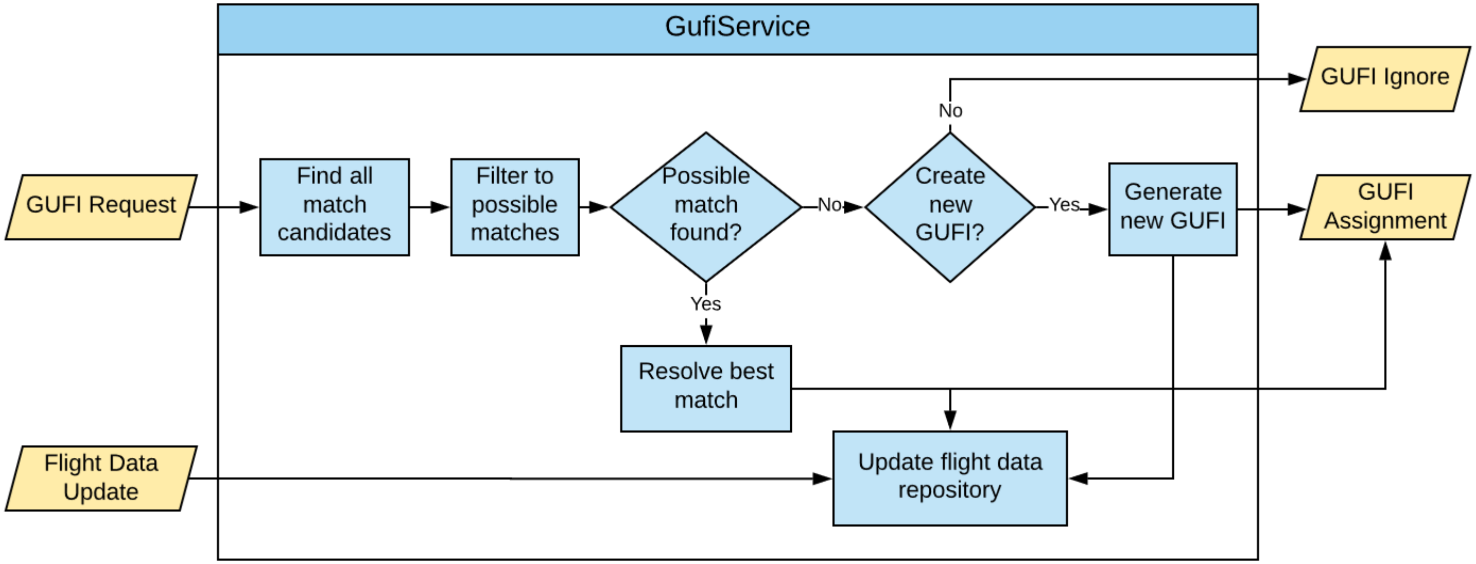
- ATD2 DFW GufiService
  - Handles GUFI requests from over 10 different data sources.
  - Stores the flight data state of each GUFI.
- External data feed processors
  - Request GUFI from GufiService.
  - Messages to Fuser include GUFI.
- GufiService performance (DFW Fuser):
  - Total:
    - ~15M avg messages handled per day. ~170 per second.
    - ~110K avg GUFIs managed per day.
    - ~200 avg messages per GUFI.
  - DFW:
    - ~3.5M avg messages handled per day.
    - ~7,500 avg GUFIs managed per day.
    - ~1,000 avg messages per GUFI.



Flight Matching: ATD2 Approach

- Basic approach to matching a flight
  - Find the best possible flight match, if one exists in the known data.
  - Otherwise, create a new GUFI, if sufficient data exists.

- Core flight matching data
  - Aircraft ID / Callsign (e.g. SWA568).
  - Origin / Destination airports (e.g. DFW -> LGA).
  - Flight time windows (e.g. 9:30 - 13:45).
  - Position (e.g. latitude / longitude / altitude).
  - External data feed system IDs (e.g. TFMData flightRef).
  - Aircraft attributes (e.g. registration number, Mode S transponder)

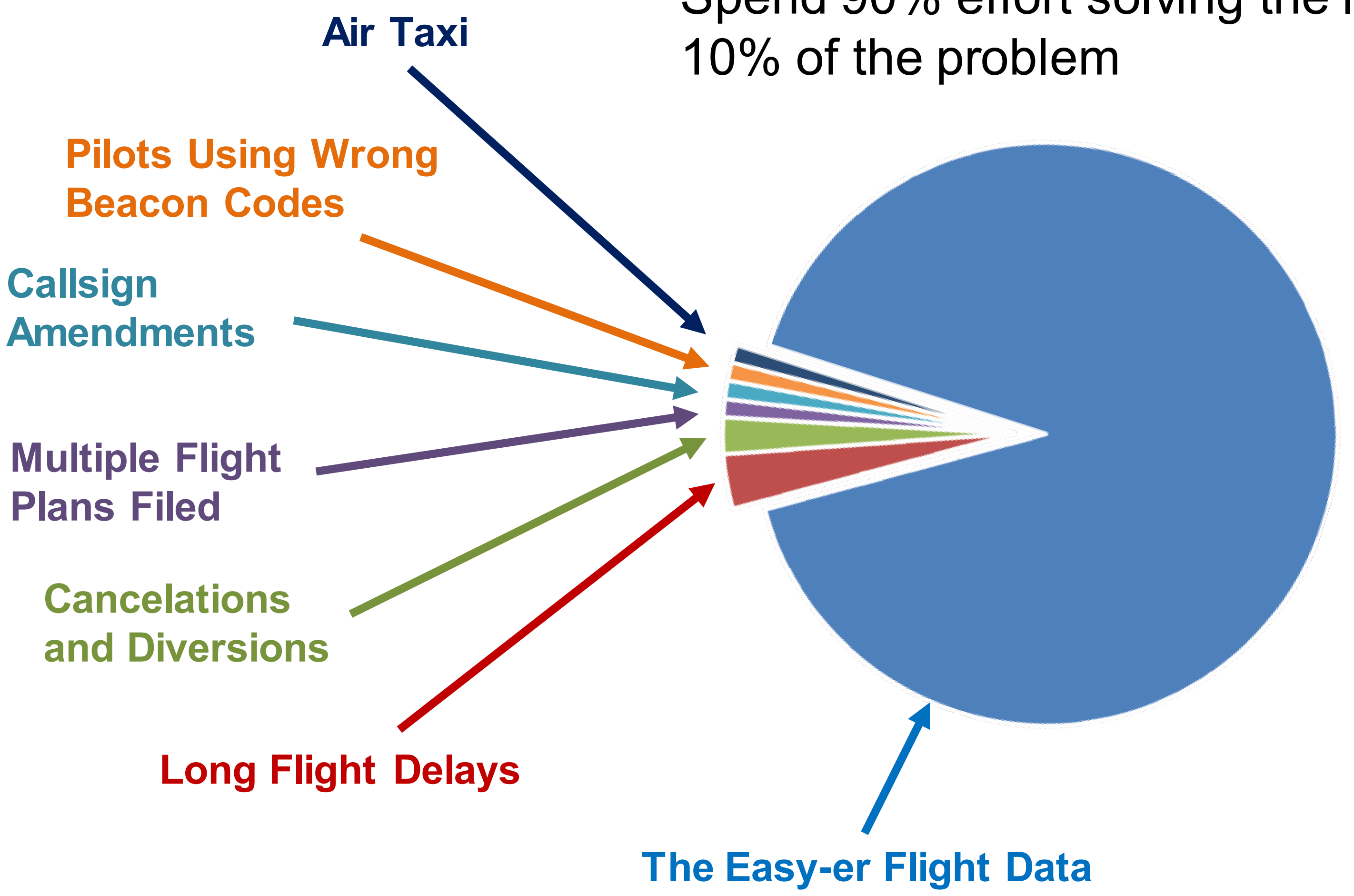


Flight Matching: Example

Time	Message	TFM FlightData	SFDPS	TBFM MIS	STDDS SMES	ATD2 GUFI
Sunday 19:00	**Scheduled flight plan** AAL045 DFW -> CLT Monday 19:00->21:59	X				AAL045.DFW.CLT.021019.1900.0000
Monday 17:00	**Filed flight plan** AAL045 DFW -> CLT Monday 19:00->21:59	X	X	X		AAL045.DFW.CLT.021019.1900.0000
Monday 17:30	**Amend flight plan route** AAL045 DFW -> CLT Monday 19:00->21:59	X	X	X		AAL045.DFW.CLT.021019.1900.0000
Monday 18:30	**CDM times update** AAL045 DFW -> CLT Monday 20:05->22:15	X				AAL045.DFW.CLT.021019.1900.0000
Monday 18:50	**EDCT** AAL045 EDCT=20:24	X		X		AAL045.DFW.CLT.021019.1900.0000
Monday 19:02	**Scheduled flight plan** AAL045 DFW -> CLT Tuesday 19:00->21:59	X				AAL045.DFW.CLT.021119.1902.0000
Monday 19:55	**Surface surveillance** AAL045 ASEX=KDFW				X	AAL045.DFW.CLT.021019.1900.0000
Monday 20:25	**Flight Departure** AAL045	X	X	X		AAL045.DFW.CLT.021019.1900.0000
Monday 20:27	**Airborne surveillance** AAL045	X	X		X	AAL045.DFW.CLT.021019.1900.0000

Flight Matching: Is Simple, Right?

Spend 90% effort solving the last 10% of the problem



Air Taxi

Pilots Using Wrong Beacon Codes

Callsign Amendments

Multiple Flight Plans Filed

Cancelations and Diversions

Long Flight Delays

The Easy-er Flight Data

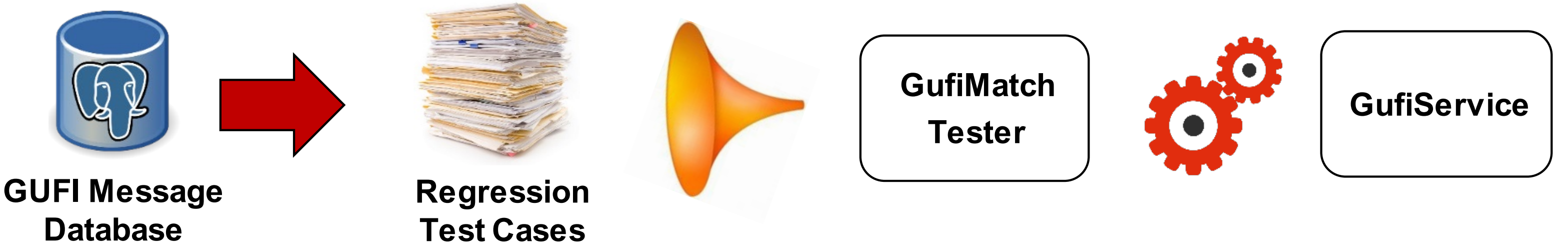
Flight Matching: Regression Testing

- Regression Testing = Key to matching logic improvement
  - Definition: A suite of tests that are required to pass when software changes are made.
  - Matching logic improvement is heuristic-based, adaptive process.
    - There is no 100% solution. Flight data is always scheming to defeat you.
  - Goal is intelligent whack-a-mole – fix one issue without causing another.

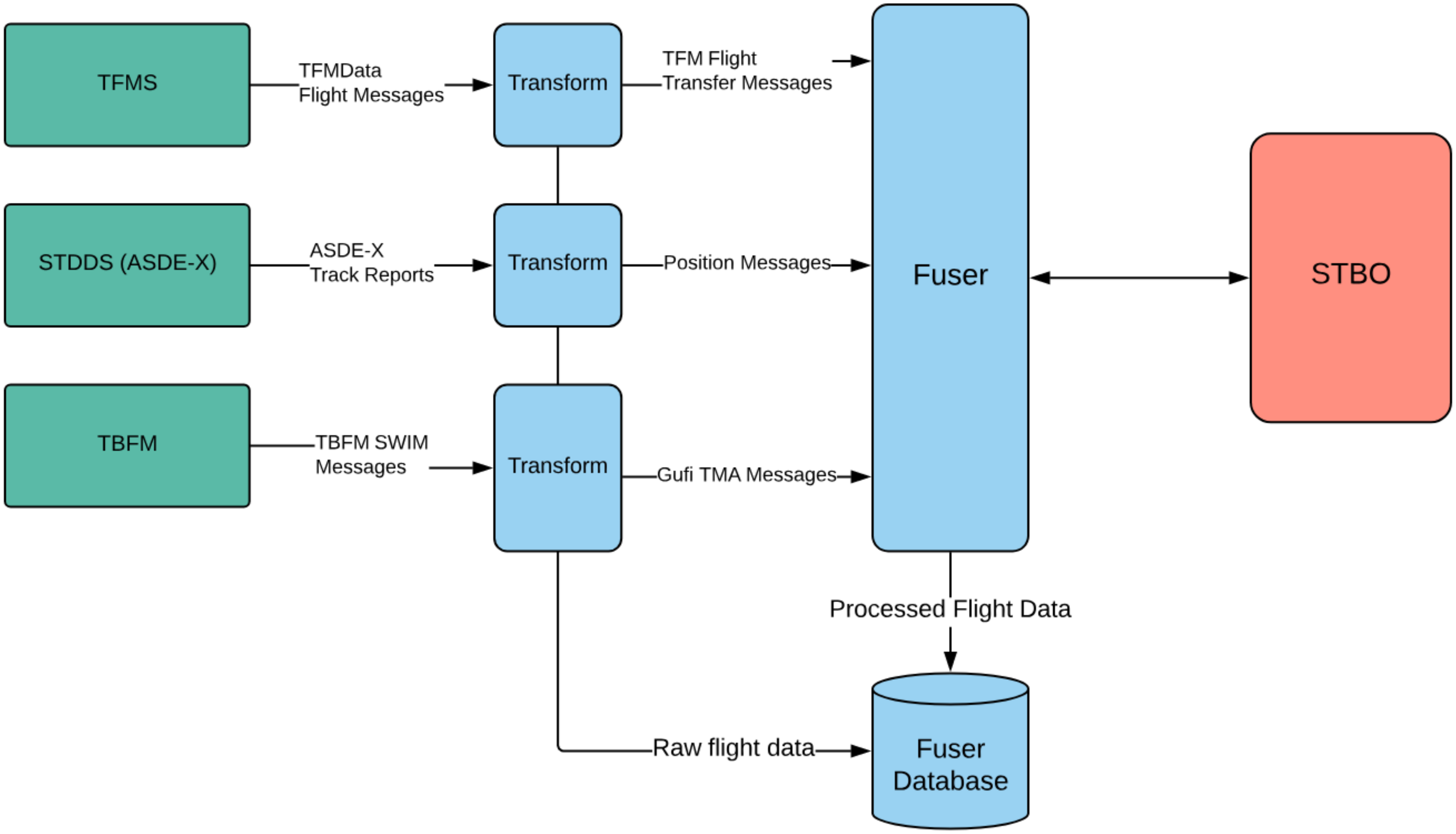
- Purpose
  - Emulate full matching process: replaying input messages from mock sub-components through the internal GufiService logic.
  - Reliable debugging of operational use cases.
  - Each test is simple format for analysts/testers to describe a matching problem.
  - Stockpile of regression tests to run against future development.

Flight Matching: Regression Testing

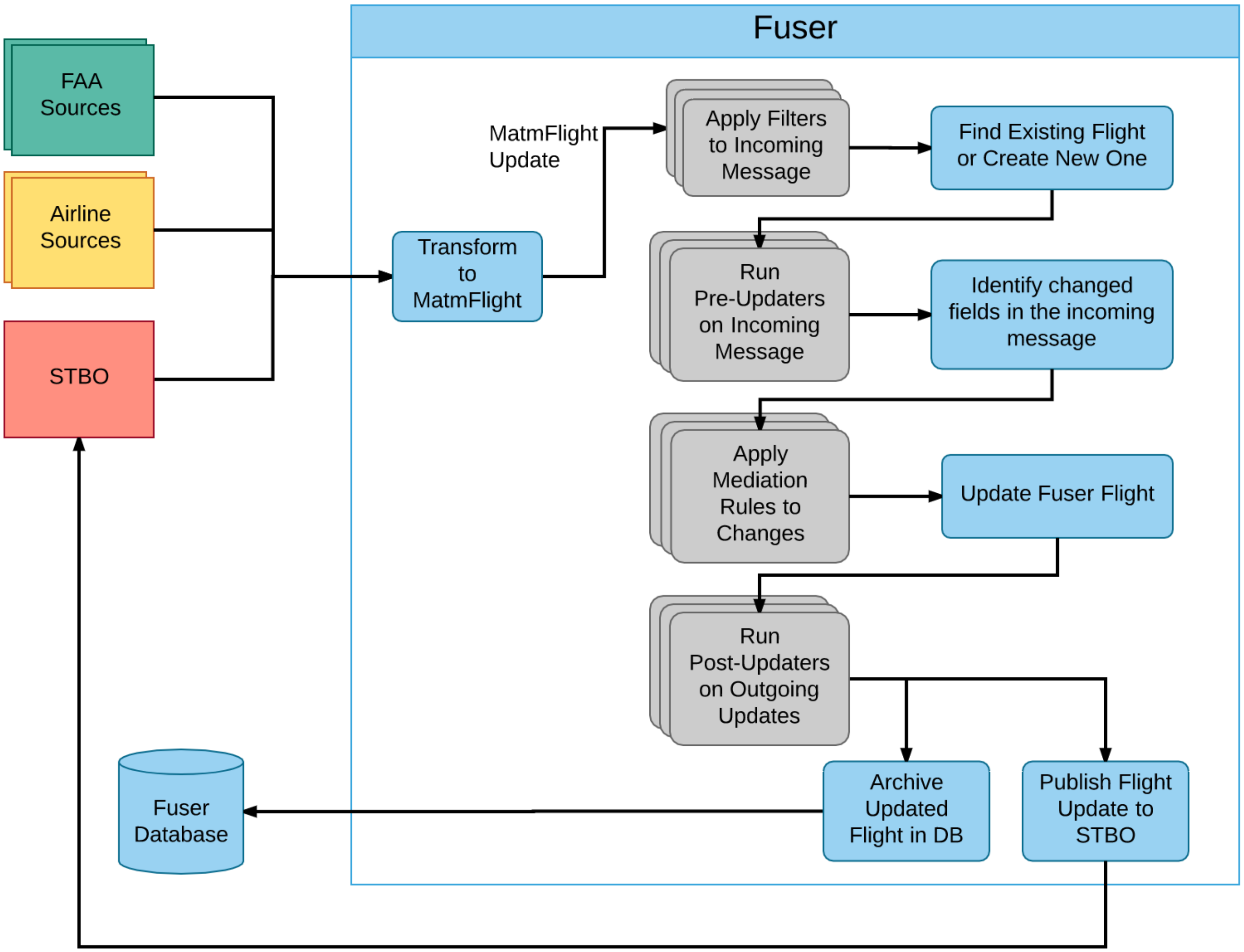
- Regression Test Process
  - Leverage database of historical GufiService messages.
  - Build match logic test cases with validated results.
  - GufiMatchTester software drives GufiService with test cases.
    - Currently over 100 regression test cases.
  - Iterate GufiService logic improvements until all tests pass.



Flight Data Transformations



Fuser Flight Processing



Fuser
FAA Sources
Airline Sources
STBO
MatmFlight Update
Transform to MatmFlight
Apply Filters to Incoming Message
Find Existing Flight or Create New One
Run Pre-Updaters on Incoming Message
Identify changed fields in the incoming message
Apply Mediation Rules to Changes
Update Fuser Flight
Run Post-Updaters on Outgoing Updates
Archive Updated Flight in DB
Publish Flight Update to STBO
Fuser Database

Filtering

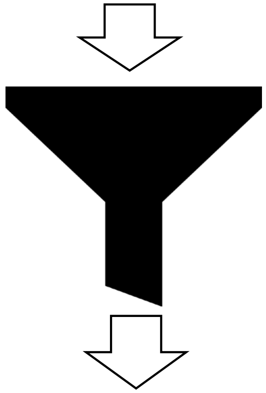
- Filtering is used to filter out an entire update before the data is applied to the fused flight
  - Eliminate unnecessary processing and/or updates that reduce data quality.
  - Filtering is based on defined rules:

Filter Name	Description	Reason
AttributeFilter	See Industry Day Documentation for more details Link.	In certain cases, we know that a source has bad data in certain fields, we don't trust the values, or we feel we will have better data from another source.
FlightPositionFilter	Filter out the position update if that latitude, longitude, or timestamp is null.	The systems needs to have all three to have a valid position. At times we are getting bad data with one of those three fields missing causing problems downstream.
GufiFilter	Filter out any messages that have not been assigned a GUFI (Global Unique Flight Identifier).	These are messages we were unable to match typically due to a lack of information.
LocationFilter	Filter out any messages that are not arriving at CLT, departing from CLT, or a Surveillance target at CLT.	In this case, the Fuser was for an STBO system operating for Charlotte Douglas International Airport (CLT) and Therefore only needed data relevant to CLT. Implemented to keep performance under control by not processing data not relevant to CLT.

Filtering Sample

acid	Departure Aerodrome	departure stand earliest time	Arrival Aerodrome	Last update source	System id	Timestamp
ABC1234	CLT	2017-04-05 11:00	DFW	TFM_TFDM	ABC	2017-04-05 10:00
ABC4567	ATL	2017-04-05 11:15	ORD	TFM_TFDM	ABC	2017-04-05 10:00
ABC8999	CLT	2017-04-05 11:30	JFK	TFM_TFDM	ABC	2017-04-05 10:00

Not a flight operating at CLT



acid	Departure Aerodrome	departure stand earliest time	Arrival Aerodrome	Last update source	System id	Timestamp
ABC1234	CLT	2017-04-05 11:00	DFW	TFM_TFDM	ABC	2017-04-05 10:00
ABC8999	CLT	2017-04-05 11:30	JFK	TFM_TFDM	ABC	2017-04-05 10:00

Attribute Filtering

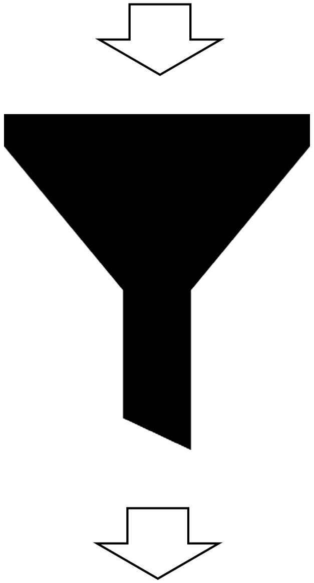
- Attribute filtering is used to filter out fields before they are applied to the fused flight, based on the defined rules

- Attribute Filtering is used when
  - A source is known to have bad or untrustworthy data in certain fields
  - Better data is likely to exist in another source for certain fields

Field	Excluded by Sources	Reason
arrivalFixActualTime	Airline Data, 3rd Party Data	Relying on STBO detection times for the actual values
arrivalMovementAreaActualTime	Airline Data, 3rd Party Data	Relying on STBO detection times for the actual values
arrivalRunwayActualTime	Airline Data, 3rd Party Data, TFMS_Flight, TFM_Terminal	Relying on STBO detection times for the actual values
departureFixActualTime	Airline Data, 3rd Party Data	Relying on STBO detection times for the actual values
departureMovementAreaActualTime	Airline Data, 3rd Party Data	Relying on STBO detection times for the actual values
departureRunwayActualTime	Airline Data, 3rd Party Data, TBFM, TFMS_Flight, TFMS_Terminal	Relying on STBO detection times for the actual values
departureQueueEntryActualTime	Airline Data, 3rd Party Data	Relying on STBO detection times for the actual values
departureRunwayAssigned	TBFM	The departure runway is only coming with the runway number and not an indication if is L,R, or C. For example 18 instead of 18L..

Attribute Filtering Sample

acid	departure runway assigned	Last update source	System id	Timestamp
ABC1234	CLT_36	TBFM	SWIM	2019-04-05 10:00



Departure Runway From TBFM are not Reliable (flt.drw)

acid	Last update source	System id	Timestamp
ABC1234	TBFM	SWIM	2019-04-05 10:00

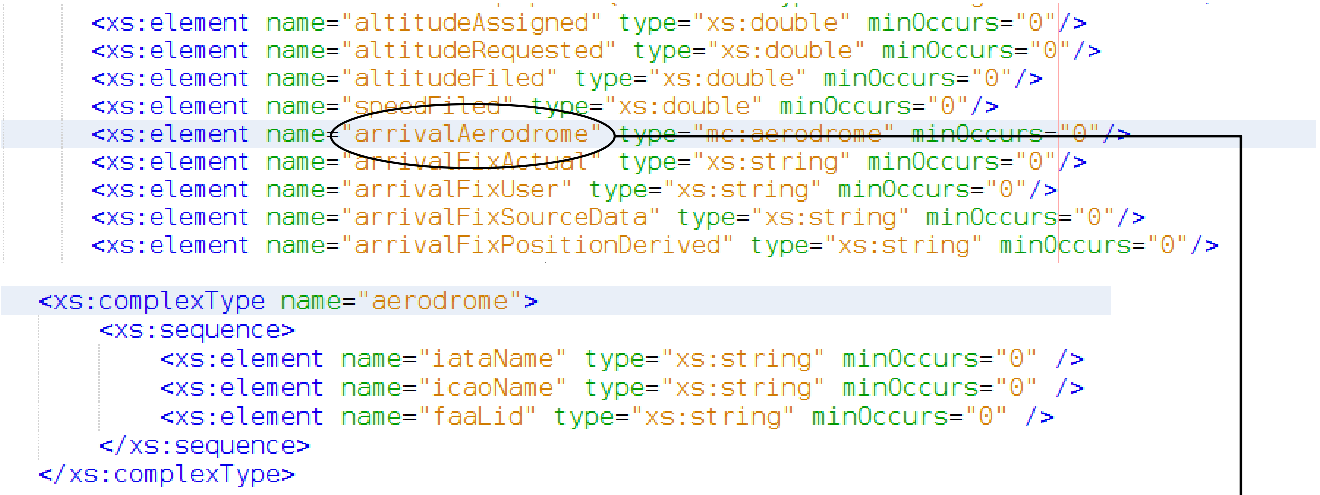
Mediation

- Fuser mediation processing
  - Input data correction
    - Data source has known errors or anomalies
  - Value-added data computations
    - Create new data elements not available in the input data source
  - Input data source priority rules
    - Used to define a precedence/authority between sources providing data for the same data elements.
    - Implemented when necessary
      - Default behavior allows any data source to modify a value
    - Filter out data modifications from one data source, if data modifications already exist from a higher priority source.
      - E.g. TFMData messages are more favorable in setting the Fuser schema “routeText” data element than TBFM MIS.
- More information
  - https://aviationsystems.arc.nasa.gov/atd2-industry-days/fuser/Data-Mediation-Overview_85328193.html

Fuser Metadata

- Fuser Metadata
  - In support of the mediation rules, Fuser tracks metadata of each Fuser schema data element:
    - Last modification input data source and message type.
    - Last modification timestamp.

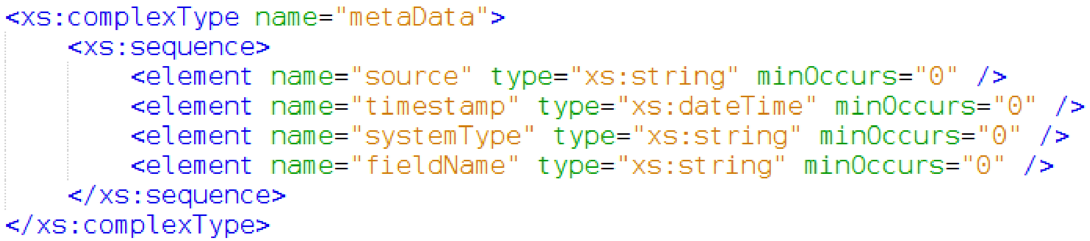
Fuser Schema



```
<xs:element name="altitudeAssigned" type="xs:double" minOccurs="0"/>
<xs:element name="altitudeRequested" type="xs:double" minOccurs="0"/>
<xs:element name="altitudeFiled" type="xs:double" minOccurs="0"/>
<xs:element name="speedFiled" type="xs:double" minOccurs="0"/>
<xs:element name="arrivalAerodrome" type="mc:aerodrome" minOccurs="0"/>
<xs:element name="arrivalFixActual" type="xs:string" minOccurs="0"/>
<xs:element name="arrivalFixUser" type="xs:string" minOccurs="0"/>
<xs:element name="arrivalFixSourceData" type="xs:string" minOccurs="0"/>
<xs:element name="arrivalFixPositionDerived" type="xs:string" minOccurs="0"/>

<xs:complexType name="aerodrome">
    <xs:sequence>
        <xs:element name="iataName" type="xs:string" minOccurs="0" />
        <xs:element name="icaoName" type="xs:string" minOccurs="0" />
        <xs:element name="faaLid" type="xs:string" minOccurs="0" />
    </xs:sequence>
</xs:complexType>
```

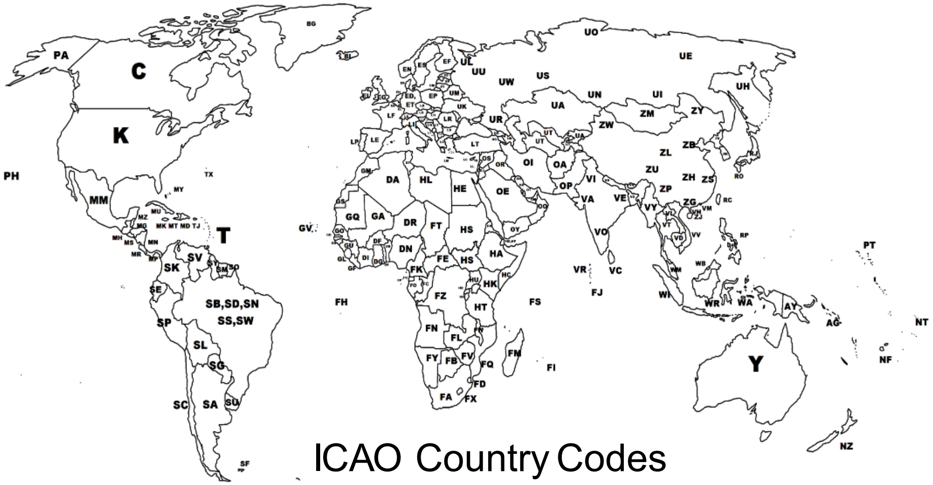
Fuser MetaData



```
<xs:complexType name="metaData">
    <xs:sequence>
        <element name="source" type="xs:string" minOccurs="0" />
        <element name="timestamp" type="xs:dateTime" minOccurs="0" />
        <element name="systemType" type="xs:string" minOccurs="0" />
        <element name="fieldName" type="xs:string" minOccurs="0" />
    </xs:sequence>
</xs:complexType>
```

Fuser Airport Mediation Use Case

- ICAO vs IATA
  - ICAO = International Civil Aviation Organization
  - IATA = International Air Transport Association
  - These organizations don't agree on how to name things.
    - FAA generally uses IDs similar to ICAO.
    - Airlines generally prefer IATA.

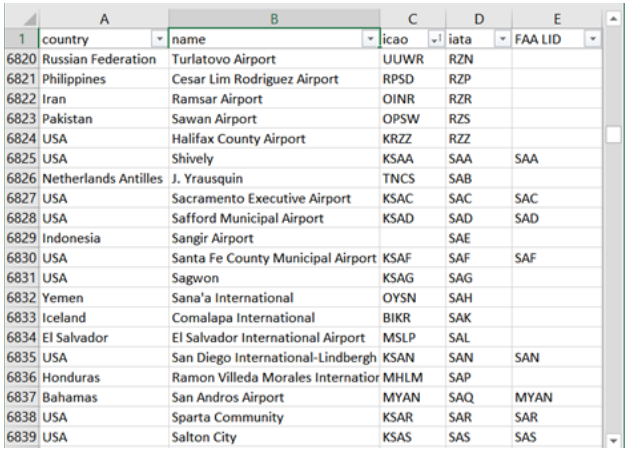


ICAO Country Codes

	IATA	ICAO	FAA
Dallas Love Field Airport	DAL	KDAL	DAL
Hilton Head Airport	HHH	KHXD	HXD
Ardmore Downtown Executive Airport	AHD		1F0
Gastonia Municipal Airport		KAKH	AKH
Augusta Municipal Airport			3AU
Boeing 737-700 Aircraft Type	73G	B737	B737
Air Carrier	AA	AAL	AAL

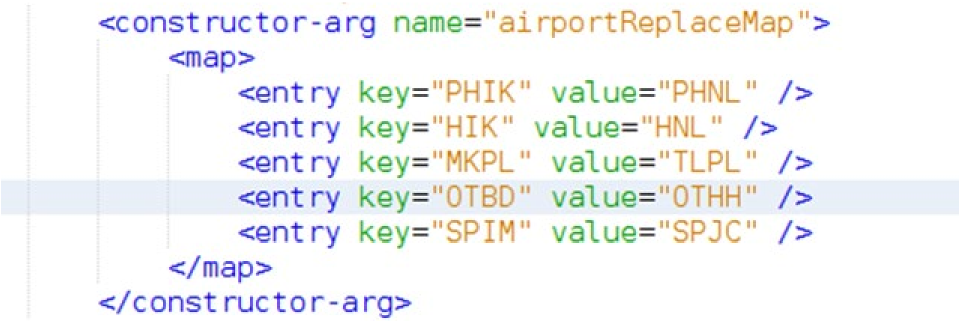
Fuser Airport Mediation Use Case

- Airport Conversion and Correction by Source
  - Mapping over 27k ICAO / IATA / FAA LID.
  - Based upon input data source, set ICAO, IATA, and FAA LID values for all messages.



	A	B	C	D	E
1	country	name	icao	iata	FAA LID
6820	Russian Federation	Turlatovo Airport	UUWR	RZN	
6821	Philippines	Cesar Lim Rodriguez Airport	RPSD	RZP	
6822	Iran	Ramsar Airport	OINR	RZR	
6823	Pakistan	Sawan Airport	OPSW	RZS	
6824	USA	Halifax County Airport	KRZZ	RZZ	
6825	USA	Shively	KSAA	SAA	SAA
6826	Netherlands Antilles	J. Yrausquin	TNCS	SAB	
6827	USA	Sacramento Executive Airport	KSAC	SAC	SAC
6828	USA	Safford Municipal Airport	KSAD	SAD	SAD
6829	Indonesia	Sangir Airport		SAE	
6830	USA	Santa Fe County Municipal Airport	KSAF	SAF	SAF
6831	USA	Sagwon	KSAG	SAG	
6832	Yemen	Sana'a International	OYSN	SAH	
6833	Iceland	Comalapa International	BIKR	SAK	
6834	El Salvador	El Salvador International Airport	MSLP	SAL	
6835	USA	San Diego International-Lindbergh	KSAN	SAN	SAN
6836	Honduras	Ramon Villeda Morales Internatior	MHLM	SAP	
6837	Bahamas	San Andros Airport	MYAN	SAQ	MYAN
6838	USA	Sparta Community	KSAR	SAR	SAR
6839	USA	Salton City	KSAS	SAS	SAS

- Special Cases and Data Anomalies
  - TBFM MIS “airport” can be a waypoint.
    - Airborne IFR filed flight plans.
  - TFM Schedule flight plan OAG errors.
    - E.g. Doha OTBD airport closed in 2014 (replaced by OTHH), but is published in TFMS Scheduled flight plan messages.

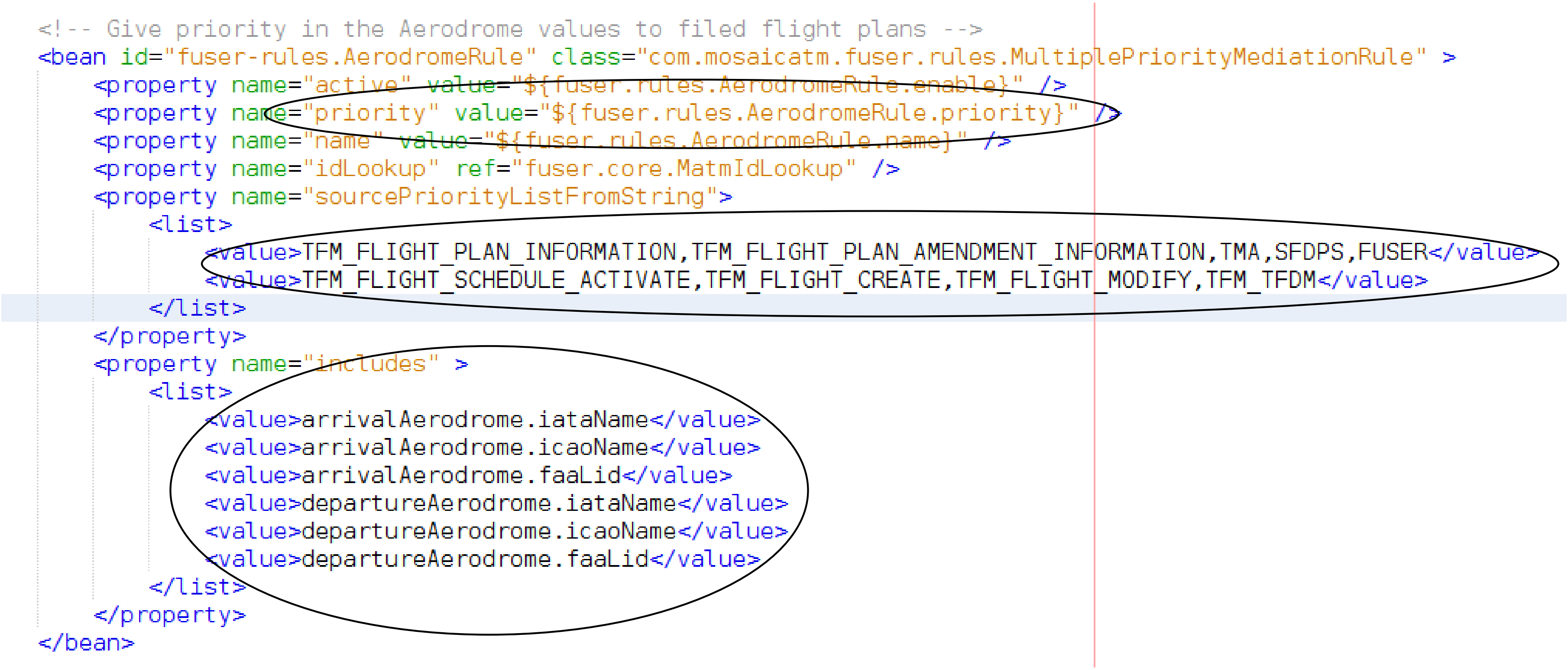


```
<constructor-arg name="airportReplaceMap">
    <map>
        <entry key="PHIK" value="PHNL" />
        <entry key="HIK" value="HNL" />
        <entry key="MKPL" value="TLPL" />
        <entry key="OTBD" value="OTHH" />
        <entry key="SPIM" value="SPJC" />
    </map>
</constructor-arg>
```

Fuser Airport Mediation Use Case

- Fuser Airport Mediation Rule
  - Problem: When flights amend the arrival airport, some TFM messages continue to use the old arrival airport.
    - Without mediation, the arrival airport toggles between the current and old values.
  - Solution: Fuser mediation to ignore data modifications by TFM sources that may use the incorrect arrival airport.

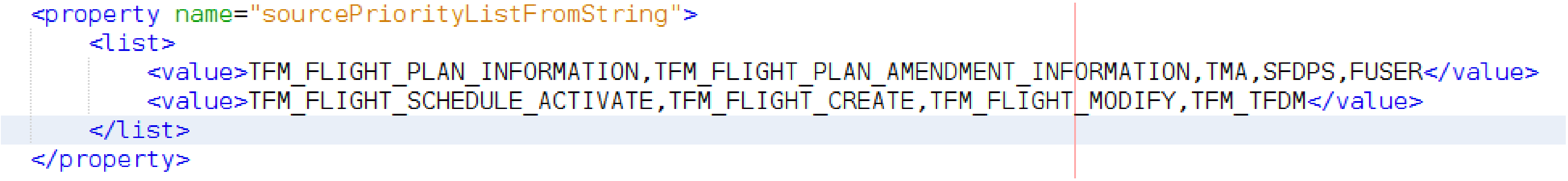
Fuser Mediation Rule



```
<!-- Give priority in the Aerodrome values to filed flight plans -->
<bean id="fuser-rules.AerodromeRule" class="com.mosaicatm.fuser.rules.MultiplePriorityMediationRule" >
    <property name="active" value="${fuser.rules.AerodromeRule.enable}" />
    <property name="priority" value="${fuser.rules.AerodromeRule.priority}" />
    <property name="name" value="${fuser.rules.AerodromeRule.name}" />
    <property name="idLookup" ref="fuser.core.MatmIdLookup" />
    <property name="sourcePriorityListFromString">
        <list>
            <value>TFM_FLIGHT_PLAN_INFORMATION,TFM_FLIGHT_PLAN_AMENDMENT_INFORMATION,TMA,SFDPS,FUSER</value>
            <value>TFM_FLIGHT_SCHEDULE_ACTIVATE,TFM_FLIGHT_CREATE,TFM_FLIGHT_MODIFY,TFM_TFDM</value>
        </list>
    </property>
    <property name="includes" >
        <list>
            <value>arrivalAerodrome.iataName</value>
            <value>arrivalAerodrome.icaoName</value>
            <value>arrivalAerodrome.faaLid</value>
            <value>departureAerodrome.iataName</value>
            <value>departureAerodrome.icaoName</value>
            <value>departureAerodrome.faaLid</value>
        </list>
    </property>
</bean>
```

Fuser Airport Mediation Use Case

- Arrival Airport Mediation Messaging Example.



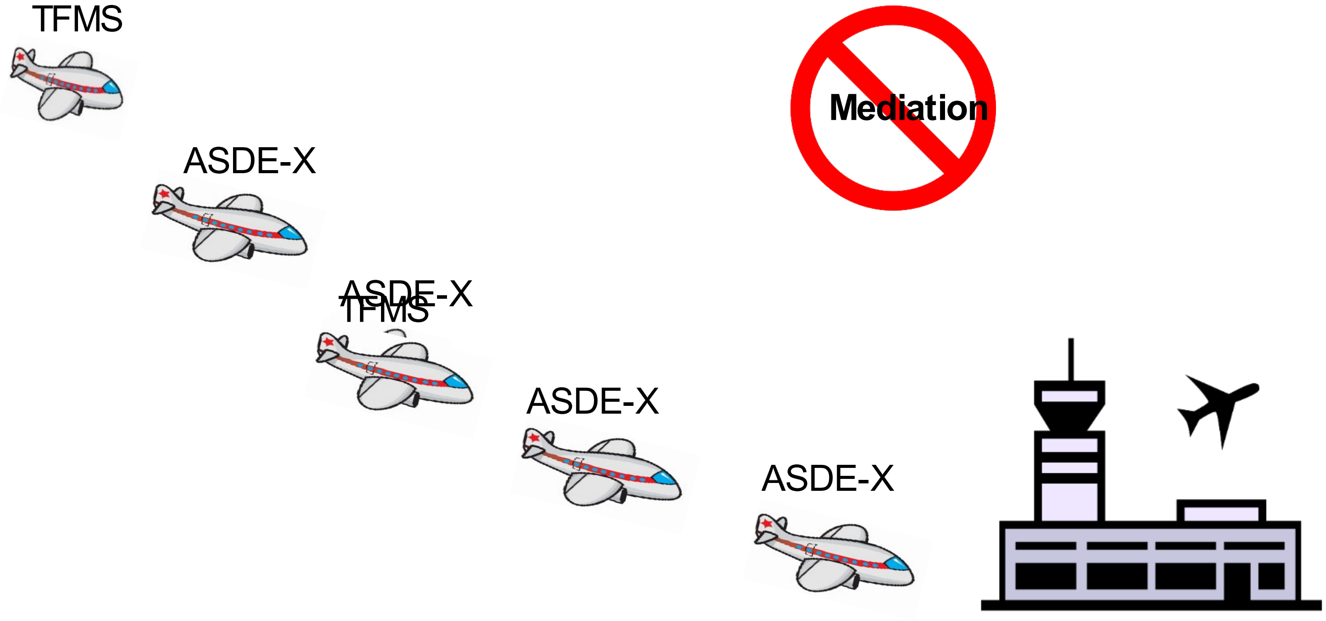
```
<property name="sourcePriorityListFromString">
    <list>
        <value>TFM_FLIGHT_PLAN_INFORMATION,TFM_FLIGHT_PLAN_AMENDMENT_INFORMATION,TMA,SFDPS,FUSER</value>
        <value>TFM_FLIGHT_SCHEDULE_ACTIVATE,TFM_FLIGHT_CREATE,TFM_FLIGHT_MODIFY,TFM_TFDM</value>
    </list>
</property>
```

Input Source				Fuser			
Time	Source	Arrival Airport		IATA	ICAO	FAA	MetaData
1500	TFMData Schedule	OTBD (error)		DOH	OTHH	OTHH	1500 (TFMData Schedule)
0900	Airline Source	DOH		DOH	OTHH	OTHH	1500 (TFMData Schedule)
1100	TFMData Filed Flight Plan	OTHH		DOH	OTHH	OTHH	1100 (TFMData Filed Flight Plan)
1130	TFMData Amend Flight Plan	OMDB		DXB	OMDB	OMDB	1130 (TFMData Amend Flight Plan)
1150	TFMData Flight Modify	OTHH		DXB	OMDB	OMDB	1130 (TFMData Amend Flight Plan)

Mediation Position Data

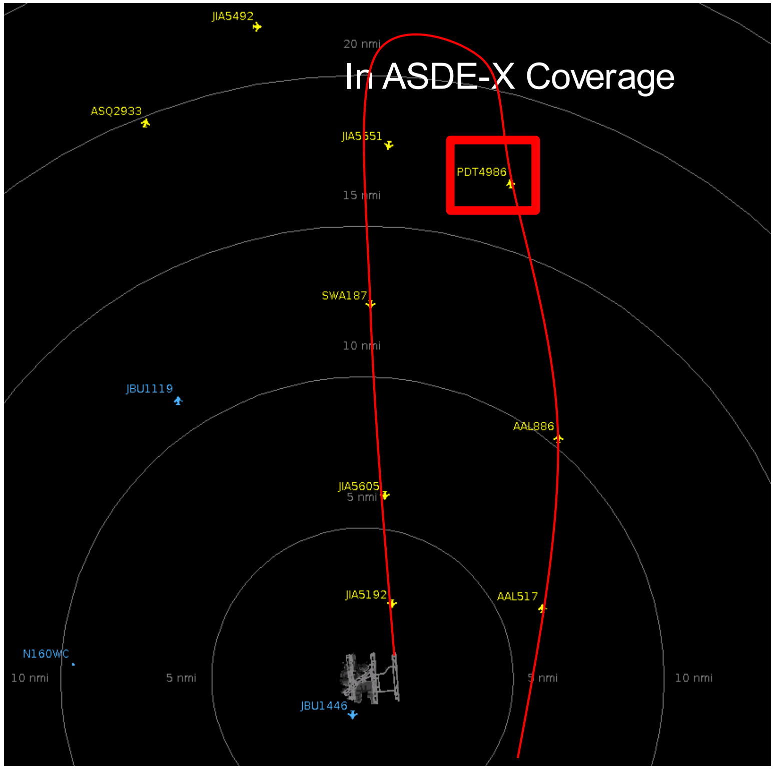
Position Data Jumpiness

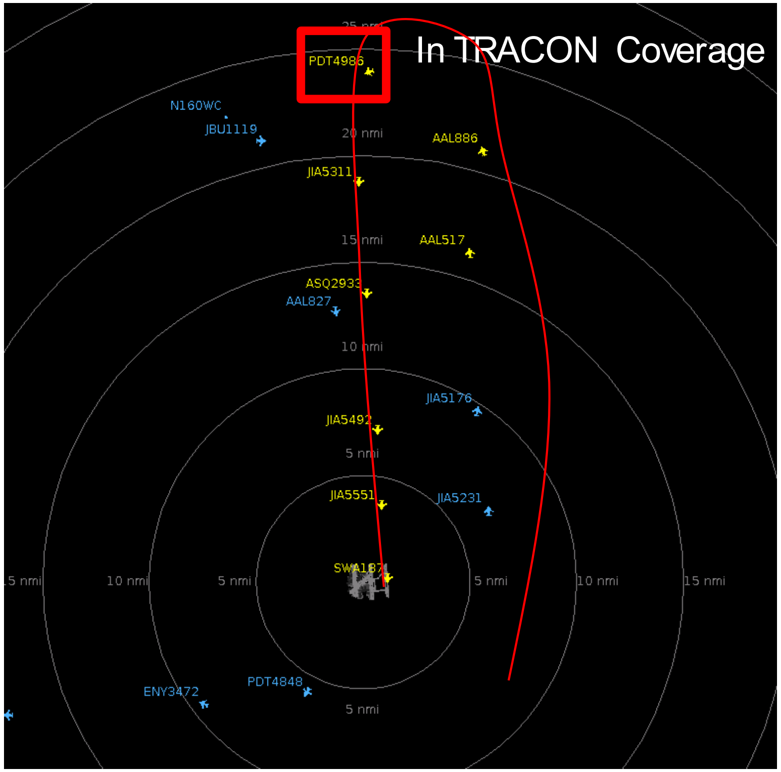
- Position data can come from numerous feeds
- If you combine the feeds without mediation, jumpiness in the data will occur



Position Coverage Transition

- The images below show the path of a flight in the terminal airspace
- In this case the flight transition from TRACON coverage to ASDE-X back to TRACON and finally back to ASDE-X


In ASDE-X Coverage


In TRACON Coverage

Mediation Example (Position Data)

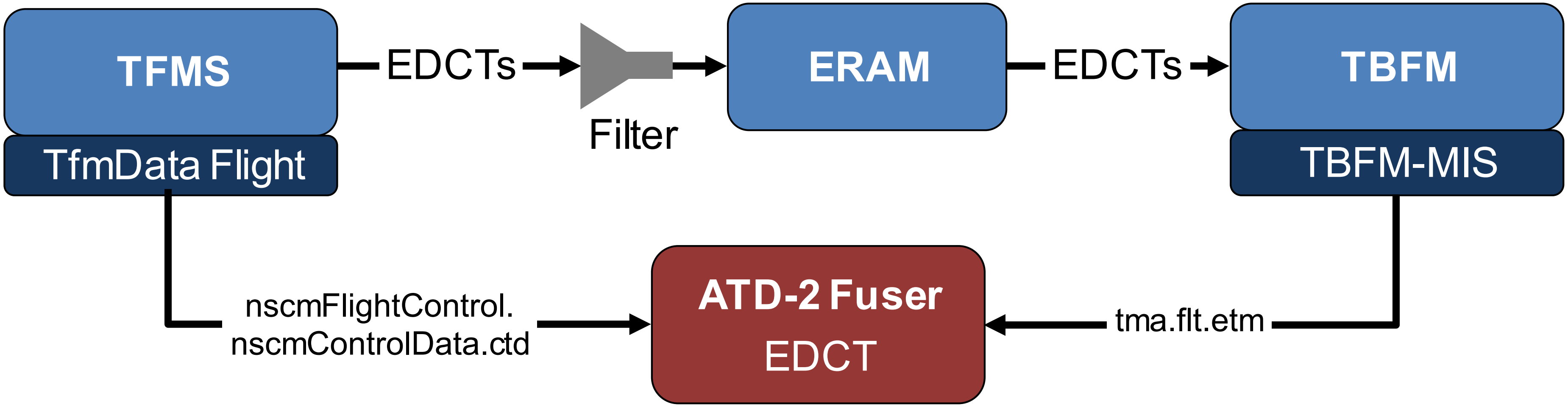
- Mediate to avoid jumpiness in the display and data
  - Define priority
  - Define a timeout
    - Transition back to a lower priority source if we quit getting data

Source	Frequency	Coverage	Priority	Timeout
STDDS (ASDE-X)	1 second	Surface to about 16 miles	1	5 seconds
STDDS TAIS	6 seconds	TRACON	2	30 seconds
SFDPS	12 seconds	NAS by CENTER stops in TRACON	3	60 seconds
TFMS	60 seconds	NAS stops in TRACON	4	60 seconds

Mediation Example EDCTs

- ATD-2 needs EDCTs for common situational awareness and runway predictions

- ATD-2 mediates TFMS and TBFM as equal sources
  - ATD-2 does not have to track filtered EDCTs separately from unfiltered



*TBFM SYNC messages had to be handled as syncs and not updates
Prevented resetting the EDCT back to an old value

Multiple Flight Plans

ATD-2 Data Elements Tracked per Flight Plan

- ATD-2 tracks individual flight plans for AEFS integration

- ATD-2 manages a list of pre-departure flight plans
  - Tracks when each flight plan was updated
  - Tracks when a flight plan is cancelled
  - Most recently updated, non-canceled, flight plan is used as the current plan for surface modeling and scheduling

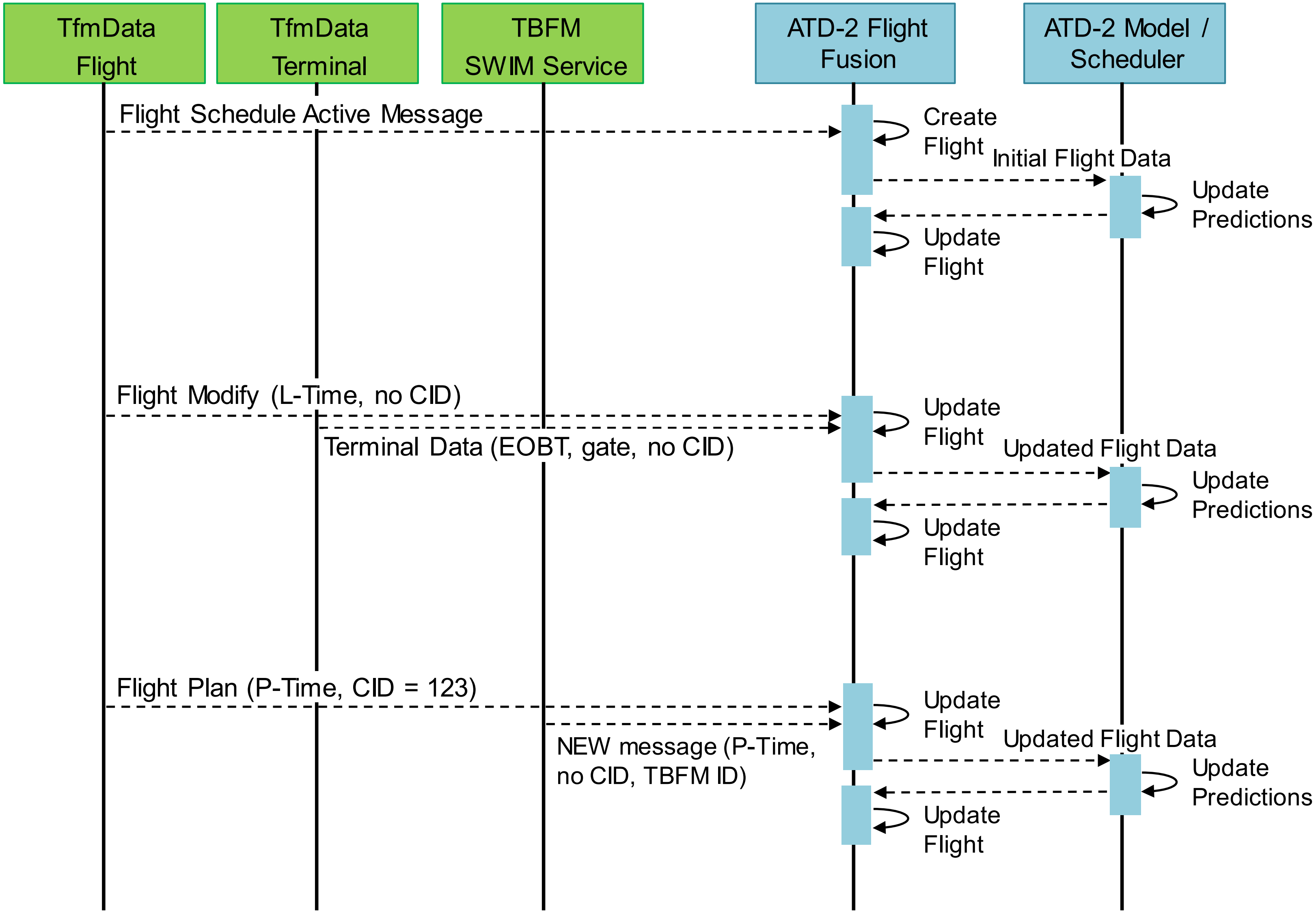
- The flight plan specific data elements are tracked per flight plan
  - P-Time, Route, Filed altitude, CID, etc.

- All other data elements are stored for the entire flight regardless of flight plan
  - L-Time, EOBT, TOBT, AOBT, Flight state, EDCT, aircraft position, etc.

- Currently uses TfmData as authoritative source for flight plan status and updates

Multiple Flight Plans

(initially the same as single flight plan use case)



TfmData Flight

TfmData Terminal

TBFM SWIM Service

ATD-2 Flight Fusion

ATD-2 Model / Scheduler

Flight Schedule Active Message

Create Flight

Initial Flight Data

Update Predictions

Update Flight

GUFI: ABC123
IOBT: 1200
ETOT: 1215

Flight Modify (L-Time, no CID)

Terminal Data (EOBT, gate, no CID)

Update Flight

Updated Flight Data

Update Predictions

Update Flight

GUFI: ABC123
IOBT:1200
L-Time: 1210
EOBT:1210
ETOT: **1225**

Flight Plan (P-Time, CID = 123)

NEW message (P-Time, no CID, TBFM ID)

Update Flight

Updated Flight Data

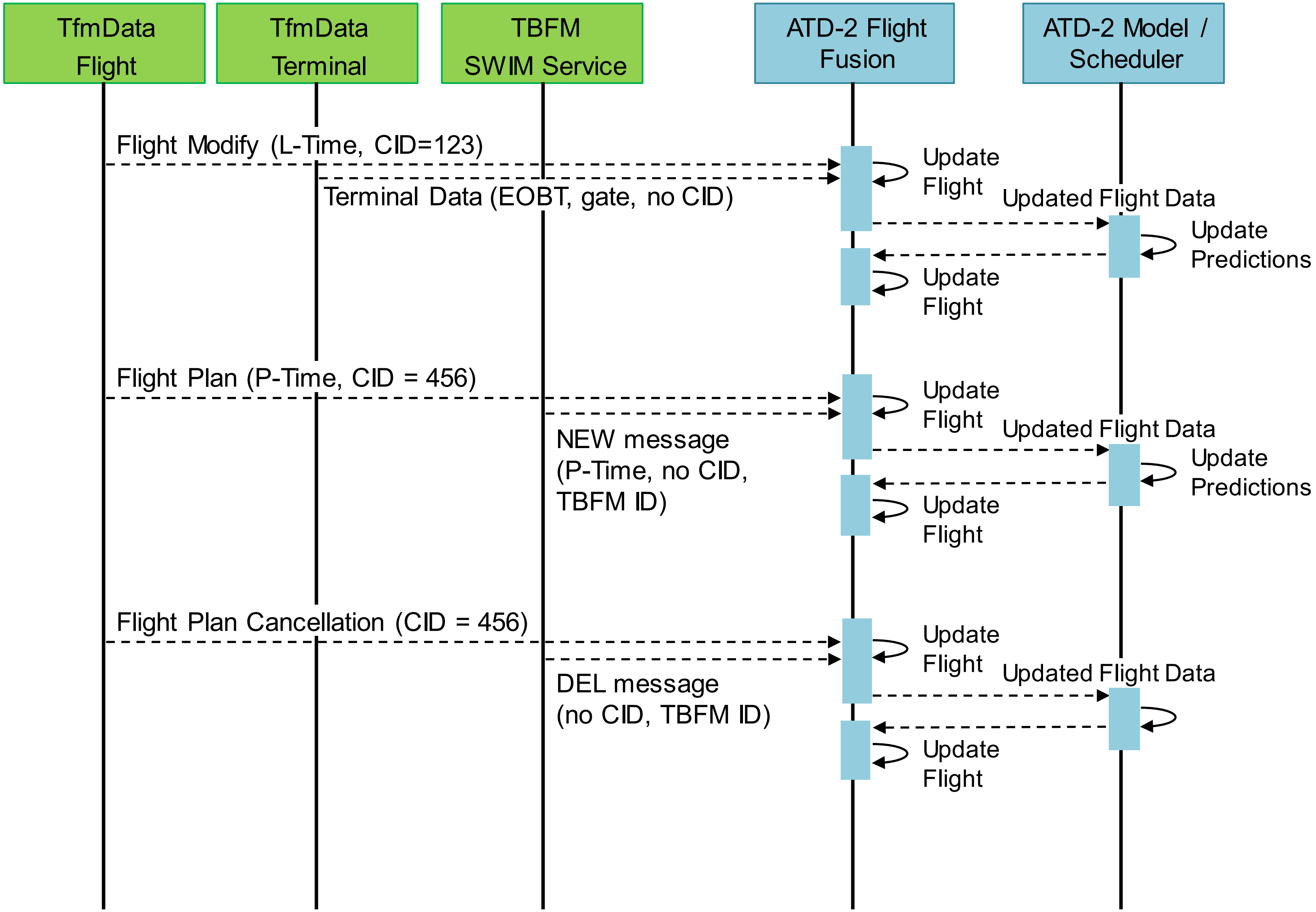
Update Predictions

Update Flight

GUFI: ABC123
IOBT:1200
L-Time: 1210
EOBT:1210
ETOT: 1225
Flight Plan 1:

- **Current = true**
- **P-Time:1210**
- **CID: 123**

Multiple Flight Plans (continued)



GUFI: ABC123
IOBT:1200
L-Time: **1230**
EOBT: **1230**
ETOT: **1245**
Flight Plan 1:
- Current = true
- P-Time:1210
- CID: 123

GUFI: ABC123
IOBT:1200
L-Time: 1230
EOBT: 1230
ETOT: 1245
Flight Plan 1:
- Current = **false**
- P-Time:1210
- CID: 123

Flight Plan 2:
- **Current = true**
- **P-Time: 1230**
- **CID: 456**

GUFI: ABC123
IOBT:1200
L-Time: 1230
EOBT: 1230
ETOT: 1245
Flight Plan 1:
- Current = **true**
- P-Time:1210
- CID: 123

~~Flight Plan 2:~~
- ~~Current = true~~
- ~~P-Time: 1230~~
- ~~CID: 456~~

Fuser Flight Data

Overview of Fuser Flight Model

- Standard naming convention used with most data elements in the Fuser Flight
- Naming convention is a based on a flattened version of the Flight Object Data Dictionary (FODD) and FIXM Schema.
- Named so that related elements are close together alphabetically
- Self documenting for the most part
- Each name consists of three parts
  - Information category
  - Resource Type
  - Source Type
- Naming convention: category_resourceType_sourceType
- Examples:
  - arrival_runway_actual_time
  - departure_spot_predicted

Prefix: Information Category

Prefix	Description
aircraft_*	Information about the physical airframe operating the flight
arrival_*	Information about the arrival portion of the flight
departure_*	Information about the departure portion of the flight
position_*	Information about the flight's position
release_*	Information about the APREQ negotiation process

Middle of Name: Resource Type

Resource Type	Description
*_runway_*	The data element related to the runway used by the flight
*_fix_*	The data element related to the fix used by the flight
*_stand_*	The data element related to the parking gate used by the flight
*_movement_area_*	The date element related to the time the flight enters or exits the movement area
*_spot_*	The data element that pertains to the physical location the flight enters or exits the movement area
*_queue_*	The data element that pertains to the time when the flight enters the departure runway queue

Postfix: Source Type

Source Type	Description
*_actual_time	The time the flight actually made use of the resource
*_controlled_time	The controlled time from a TFM TMI (GDP, AFP, GS)
*_earliest_time	The earliest time a flight is expected to use the resource by external sources (EOBT)
*_estimated_time	The time the flight is estimated to use the resource by external sources
*_initial_time	The first event time received for the resource
*_metered_time	The TMA-metered time (STA) that the flight will use the resource
*_proposed_time	The flight time based on the filed flight plan
*_scheduled_time	The time the flight is scheduled to operate by the airline
*_surface_metered_time	The time a flight should comply with as part of a Surface Metering Program
*_targeted_time	The time the flight is predicted to use the resource as set by the scheduler
*_undelayed_time	The unimpeded time the flight would use the resource is not constrained by external restriction

Postfix: Source Type (continued)

Source Type	Description
*_actual	The actual resource used by the flight
*_airline	The resource provided by an airline source or user entering the information in the RTC or STBO client display
*_assigned	The resource set by an FAA controller
*_position_derived	The resource derived from position data and adaptation by STBO
*_model	The resource derived from STBO modeling
*_decision_tree	The resource derived from STBO decision trees

Timestamp Columns

Column Name	Description
timestamp	Typically the timestamp of the source message
timestamp_fuser_processed	The time the fuser finished processing the message
timestamp_fuser_received	The time the fuser received the message
timestamp_source	The timestamp in the message supplied by the source system when available
timestamp_source_processed	The time the message was processed by the data parser
timestamp_source_received	The time the message was received by the data parser

Fuser Mappings

The most common flight identifying field must be consistent across sources, right?

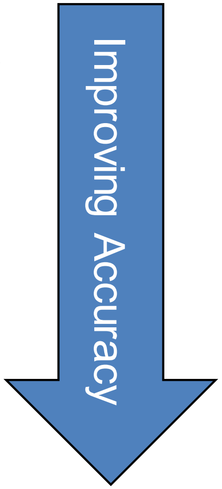
Source System	Term Used
Fuser/MatmFlight	**acid**
TFMS	qualifiedAircraftId.**aircraftId**
TBFM	tmaType.air.flt.**aid**
TFM Terminal Flight Data	**acid**
STDDS Position Report (ASDEX)	flightId.aircraftId.value
FIXM	flightIdentification.**aircraftIdentification**

Best Available Out Time

- ATD-2 departure predictions start with when the flight will leave the gate

- There are multiple data elements that provide an estimate of pushback time
  - Some data elements are available in multiple sources

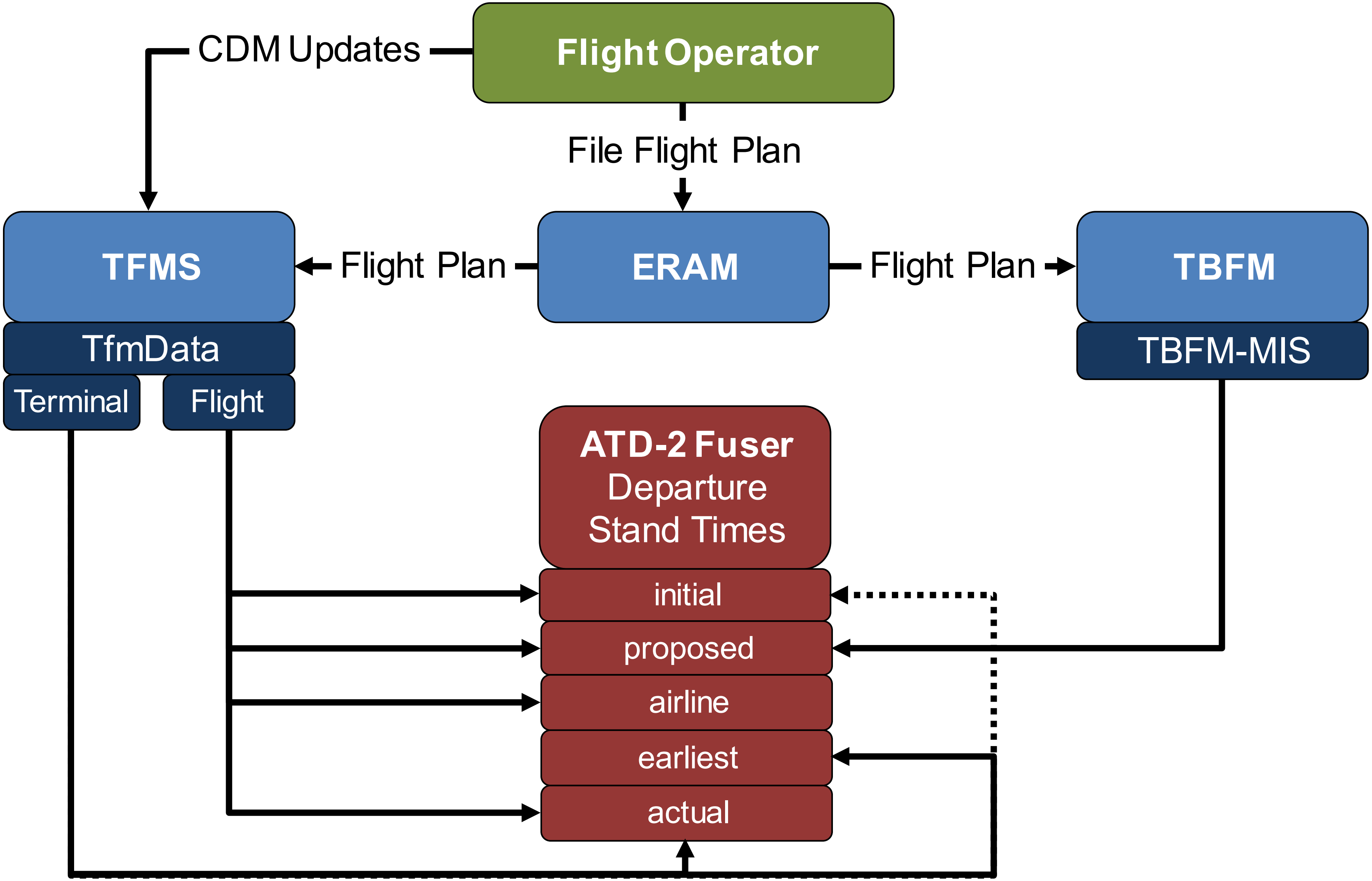
- The Fuser mediates and stores each data element individually
  - departure_stand_initial_time (IOBT / IGTD)
  - departure_stand_proposed_time (P-Time from flight plan)
  - departure_stand_airline_time (L-Time from CDM messages)
  - departure_stand_earliest_time (EOBT)
  - departure_stand_actual_time (AOBT / OUT)



Improving Accuracy

- The ATD-2 prediction engine then uses the best available data element as the start of it's predictions

Best Available Out Time



Best Available Out Time

Departure Stand Time	TfmData Flight*	TfmData Terminal	TBFM-MIS**
Initial	flight.qualifiedAircraftId.igtd	flight.departureNas. runwayDepartureTime. original.time	
Proposed	[flightPlanInformation / flightPlanAmendmentInformation]. coordinationTime.value (if coordinationPoint == departureAirport && coordinationType == PROPOSED)		tma.air.flt.ctm (if tma.air.flt.acs == PROPOSED && tma.air.flt.fps == PROPOSED && tma.air.dap == tma.air.flt.cfx)
Airline	[nscmFlightCreate / nscmFlightModify]. airlineData.flightTimeData. airlineOutTime		
Earliest		flight.departureNas. runwayDepartureTime. earliest.time	
Actual	nscmFlightModify. airlineData.flightTimeData. gateDeparture	flight.departureNas. standPositionAndTime. standTime.actual.time	

* Multiple TfmData Flight message types can contain this data. Only the most common ones are listed here.

** TBFM-MIS mapping not currently used on ATD-2 because of feedback loop between ATD-2 and TBFM

ATD-2 Fuser Mapping

- Full Fuser Mappings Available
  - https://aviationsystems.arc.nasa.gov/atd2-industry-days/fuser/Fuser-Database-Input-Mapping-Table_85328219.html

Questions

Airspace Technology Demonstration 2 (ATD-2)

ATD-2 Fuser Database

May 22, 2019

Fuser Database

- Purpose
- Database Overview
- Database Details
- Use Cases

Purpose of Databases?

Understanding the Data

What is in the raw data?

When is a specific field set in a SWIM feed?

How did a SWIM feed handle a specific flight?

End Results

How were operations yesterday / last week / last year?

Are there areas for improvement?

Did we see any benefits?

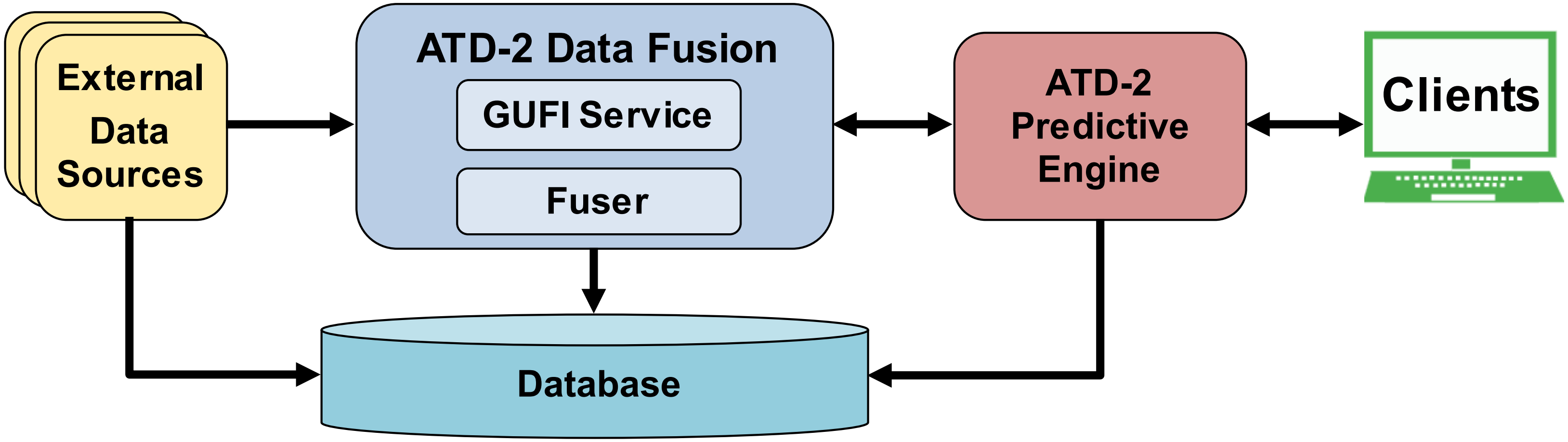
Traceability / Debugging

How did our data processing handle this case?

What triggered this update in our system?

Database Overview

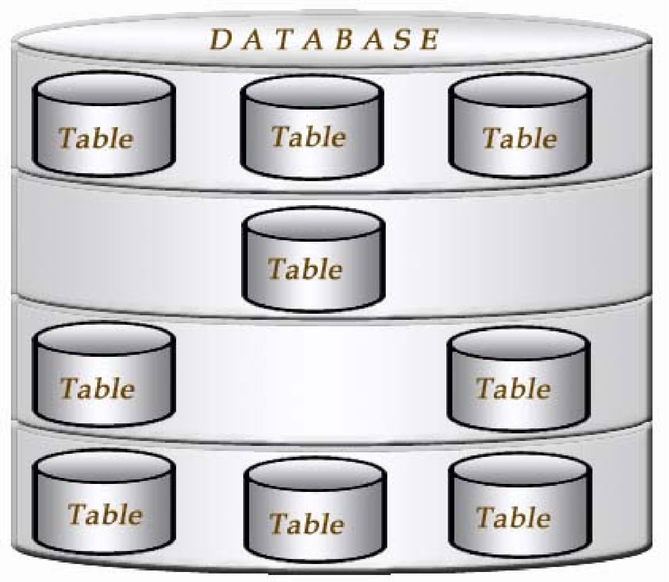
- Each ATD-2 system has a dedicated system that captures the data necessary to meet objectives
- Each database contains tables with
  - Flattened messages from external sources
  - Flight matching data
  - Fuser mediated data
  - Data from ATD-2 internal components
- Database structure prefers wide tables that contain flattened data
  - Very few joins or tree traversals needed



Database Tables

- ATD-2 Data Fusion Tables
  - Fuser flight tables
  - GUFI (Global Unique Flight Identifier) tables

- Source Data Tables
  - TfmData flight tables
  - TBFM table
  - ASDEX (STDDS SMES) table
  - TfmData Terminal table



Fuser Processed Database Tables

Table	Contains
matm_flight_all	A snapshot of the current flight state after every update
matm_flight	Just the fields that changed as part of each flight update
matm_flight_summary	A single record per flight with the current state of the flight
*_extension	Processed data updates that apply only to a specific source

Matm_Flight vs Matm_Flight_All

MATM_FLIGHT

acid	departure stand airline	departure stand airline time	departure stand actual time	Last update source	System id	Timestamp
ABC1234	B1	2017-04-05 11:00		TFM_TFDM	ABC	2017-04-05 10:00
ABC1234		2017-04-05 11:15		TFM_TFDM	ABC	2017-04-05 10:30
ABC1234			2017-04-05 11:17	TFM_TFDM	ABC	2017-04-05 11:17

MATM_FLIGHT_ALL

acid	departure stand airline	departure stand airline time	departure stand actual time	Last update source	System id	Timestamp
ABC1234	B1	2017-04-05 11:00		TFM_TFDM	ABC	2017-04-05 10:00
ABC1234	B1	2017-04-05 11:15		TFM_TFDM	ABC	2017-04-05 10:30
ABC1234	B1	2017-04-05 11:15	2017-04-05 11:17	TFM_TFDM	ABC	2017-04-05 11:17

Matm_Flight_All vs Matm_Flight_Summary

MATM_FLIGHT_ALL

acid	departure stand airline	departure stand airline time	departure stand actual time	Last update source	System id	Timestamp
ABC1234	B1	2017-04-05 11:00		TFM_TFDM	ABC	2017-04-05 10:00
ABC1234	B1	2017-04-05 11:15		TFM_TFDM	ABC	2017-04-05 10:30
ABC1234	B1	2017-04-05 11:15	2017-04-05 11:17	TFM_TFDM	ABC	2017-04-05 11:17

MATM_FLIGHT_SUMMARY – at 10:00

acid	departure stand airline	departure stand airline time	departure stand actual time	Last update source	System id	Timestamp
ABC1234	B1	2017-04-05 11:00		TFM_TFDM	ABC	2017-04-05 10:00

MATM_FLIGHT_SUMMARY – at 10:30

acid	departure stand airline	departure stand airline time	departure stand actual time	Last update source	System id	Timestamp
ABC1234	B1	2017-04-05 11:15		TFM_TFDM	ABC	2017-04-05 10:30

MATM_FLIGHT_SUMMARY – at 11:17

acid	departure stand airline	departure stand airline time	departure stand actual time	Last update source	System id	Timestamp
ABC1234	B1	2017-04-05 11:15	2017-04-05 11:17	TFM_TFDM	ABC	2017-04-05 11:17

Gufi Service (Flight Matching) Tables

Table	Contains
gufiflightmessage	Data on how each flight message from an external source was matched to a Fuser GUFI
gufiflighthistory	A snapshot of the current flight state in the GUFI flight matching service after every message

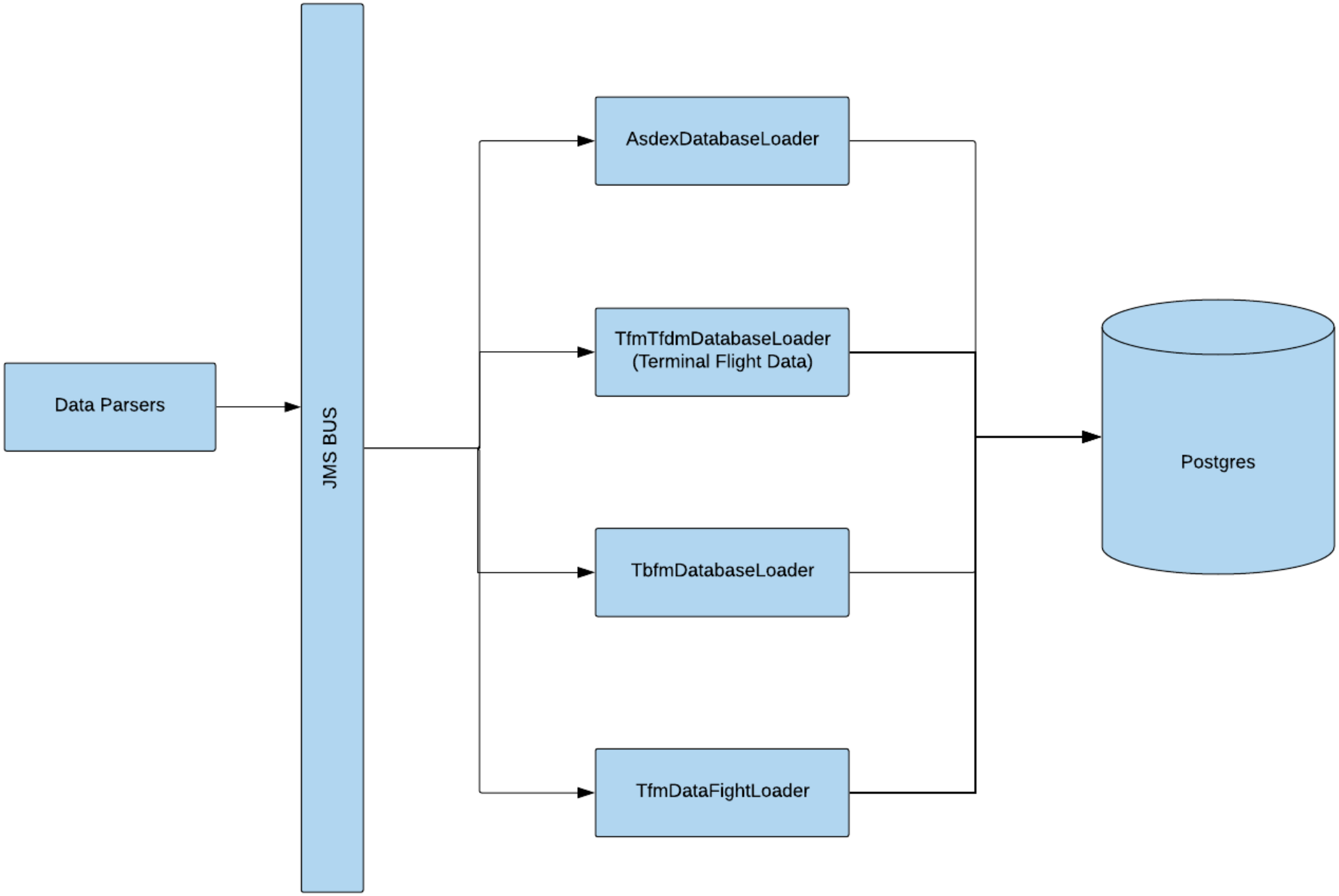
Fuser Input Message Database Tables

Table	Contains
asdex_messages	ASDE-X track data
tfm_*	TFMS flight data – one table per message type
tma_message	NASA research TBFM and SWIM TBFM data • See the source column. Contains either • SWIM • NASA-ZTL • NASA-ZDC
tfm_tfdm	TfmData Terminal Flight Data

Fuser Database – TFM Tables

TFM Table	Message Triggered When
tfm_arrival	TFMS detects that a flight has landed
tfm_boundary_crossing	A flight crosses between ARTCCs
tfm_departure	TFMS detects that a flight takes off
tfm_flight_control	TFMS issues an EDCT for a flight
tfm_flight_create	An airline submits a Flight Create message
tfm_flight_modify	An airlines submits a Flight Modify message
tfm_flight_plan	A flight plan is filed
tfm_flight_plan_amend	A flight plan is amended
tfm_flight_plan_cancel	A flight plan is canceled
tfm_flight_route	A flight's route changes
tfm_flight_schedule_activate	TFMS activates a scheduled flight 24 hours in advance
tfm_flight_times	A flight's ETD or ETA changes due to events at the origin
tfm_oceanic_report	An oceanic report is made
tfm_track	TFMS track data at 1 minute update rate

Database Loading on an ATD-2 System



Database Details

- PostgreSQL database

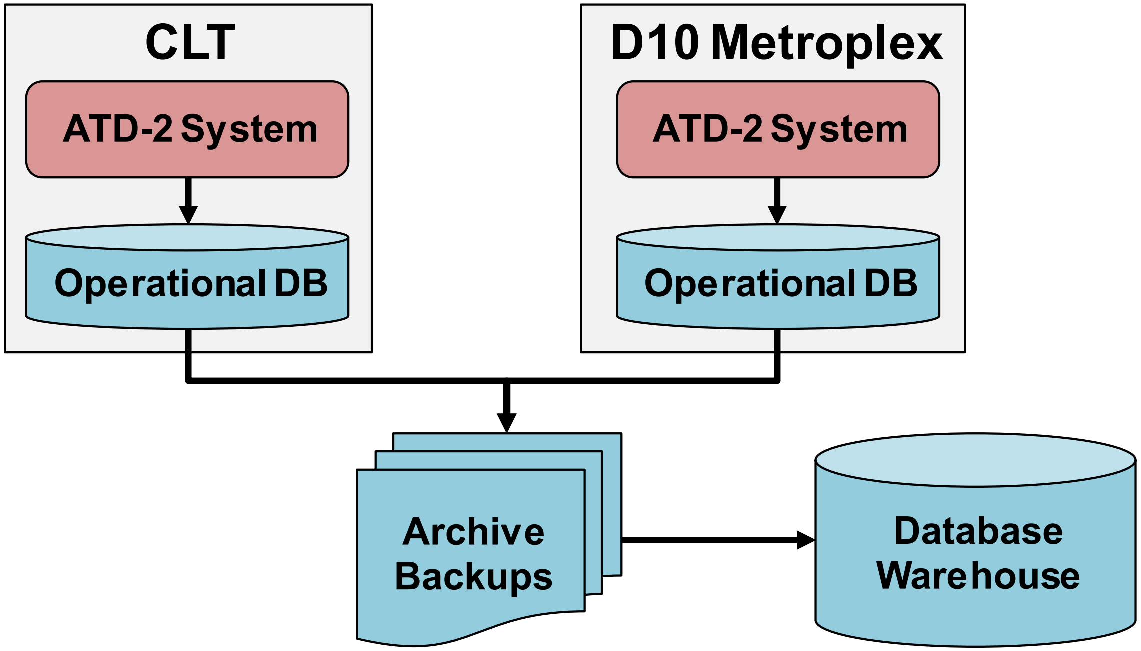
- All tables are partitioned based on timestamp
  - Improves query performance
  - Allows old data to be easily rolled off
  - Data is automatically rolled off daily
  - Data is stored long term in a warehouse

- Indexes on key data fields per table

Data Archiving

- Operational systems store 5 day's worth of data
- Warehouses store archives of data long term for post-analysis
  - Data is archived nightly to allow post-analysis scripts to run overnight

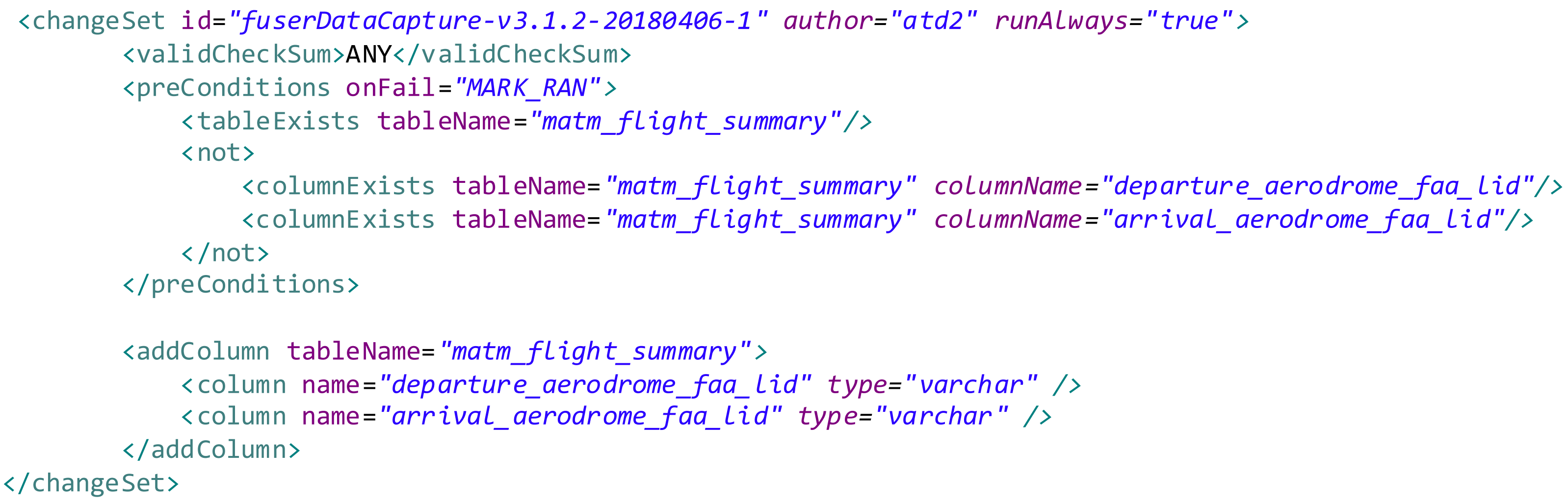
Stats	CLT Warehouse	D10 Warehouse
Flight Count	~910,000	~1,285,000
Size	~15TB	~14TB



Changes to Database Schemas

- Problem
  - Need to store new data fields with each release
  - Cannot delete data on operational system and start clean when new release is deployed
  - The new data fields will not exist in warehouse archives

- Solution
  - LiquidBase for tracking DB schema changes between versions
    - Schema changes are stored in XML
    - When new version is deployed, DB schemas are automatically updated
    - Archiving scripts read XML files and update warehouse prior to archiving new data



```xml
<changeSet id="fuserDataCapture-v3.1.2-20180406-1" author="atd2" runAlways="true">
        <validCheckSum>ANY</validCheckSum>
        <preConditions onFail="MARK_RAN">
            <tableExists tableName="matm_flight_summary"/>
            <not>
                <columnExists tableName="matm_flight_summary" columnName="departure_aerodrome_faa_lid"/>
                <columnExists tableName="matm_flight_summary" columnName="arrival_aerodrome_faa_lid"/>
            </not>
        </preConditions>

        <addColumn tableName="matm_flight_summary">
            <column name="departure_aerodrome_faa_lid" type="varchar" />
            <column name="arrival_aerodrome_faa_lid" type="varchar" />
        </addColumn>
</changeSet>
```

Use Case: Understanding the Data
Cancellations

- Question:
  - ATD-2 needed to cancel flights based on TfmData
  - TfmData has multiple triggers for cancellation
  - Which triggers should ATD-2 use?

- Approach
  - Use TfmData tables to determine the percentage of flights with a cancellation message and later had track data

- tfm_flight_plan_cancel table

acid	departure_airport	arrival_airport	flight_ref	trigger	source_timestamp	…
EDV5575	KDSM	KLGA	100000348	FD_FLIGHT_CANCEL_MSG	5/1/2019 9:12	…
UAL1965	KTPA	KEWR	99911618	HCS_CANCELLATION_MSG	5/1/2019 9:15	…
DAL146	SCEL	KATL	99972860	HCS_CANCELLATION_MSG	5/1/2019 9:16	…

Use Case: Understanding the Data
Cancellations

- Results (2019-05-01 08:00Z to 2019-05-14 08:00Z)

Trigger	Total Count	Track Count	Percent
UPDATE_INTERNATIONAL_CANCEL_TIMEOUT	79683	0	0.0%
UPDATE_CANCEL_TIMEOUT	27674	7128	25.8%
HCS_CANCELLATION_MSG	27098	15555	57.4%
FD_FLIGHT_CANCEL_MSG	11912	1009	8.5%
CANCEL_CMD	5962	75	1.3%
TMI_UPDATE	1293	178	13.8%
IADE_CANCELLATION_MSG	190	113	59.5%

- ATD-2 currently only uses FD_FLIGHT_CANCEL_MSG to mark flights as cancelled
  - HCS_CANCELLATION_MSG are used to track the cancellation of flight plans associated with a flight, but not to cancel the entire flight
  - ATD-2 has logic similar to TFMS timeout logic and so does not use timeout cancellations

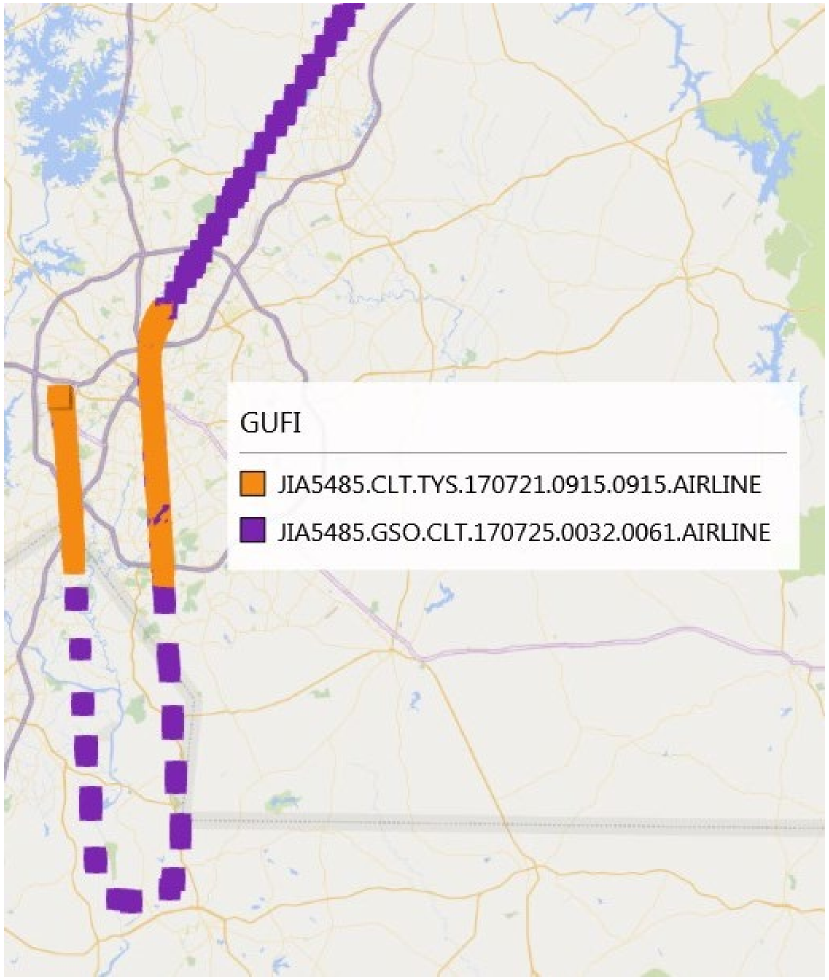
Use Case: Traceability / Debugging
Flight Matching Issues

- Question:
  - Prior to going live at CLT, there was a large push to resolve flight matching issues
  - Testers reported an issue where two aircraft icons were observed for a single arrival
  - What happened?

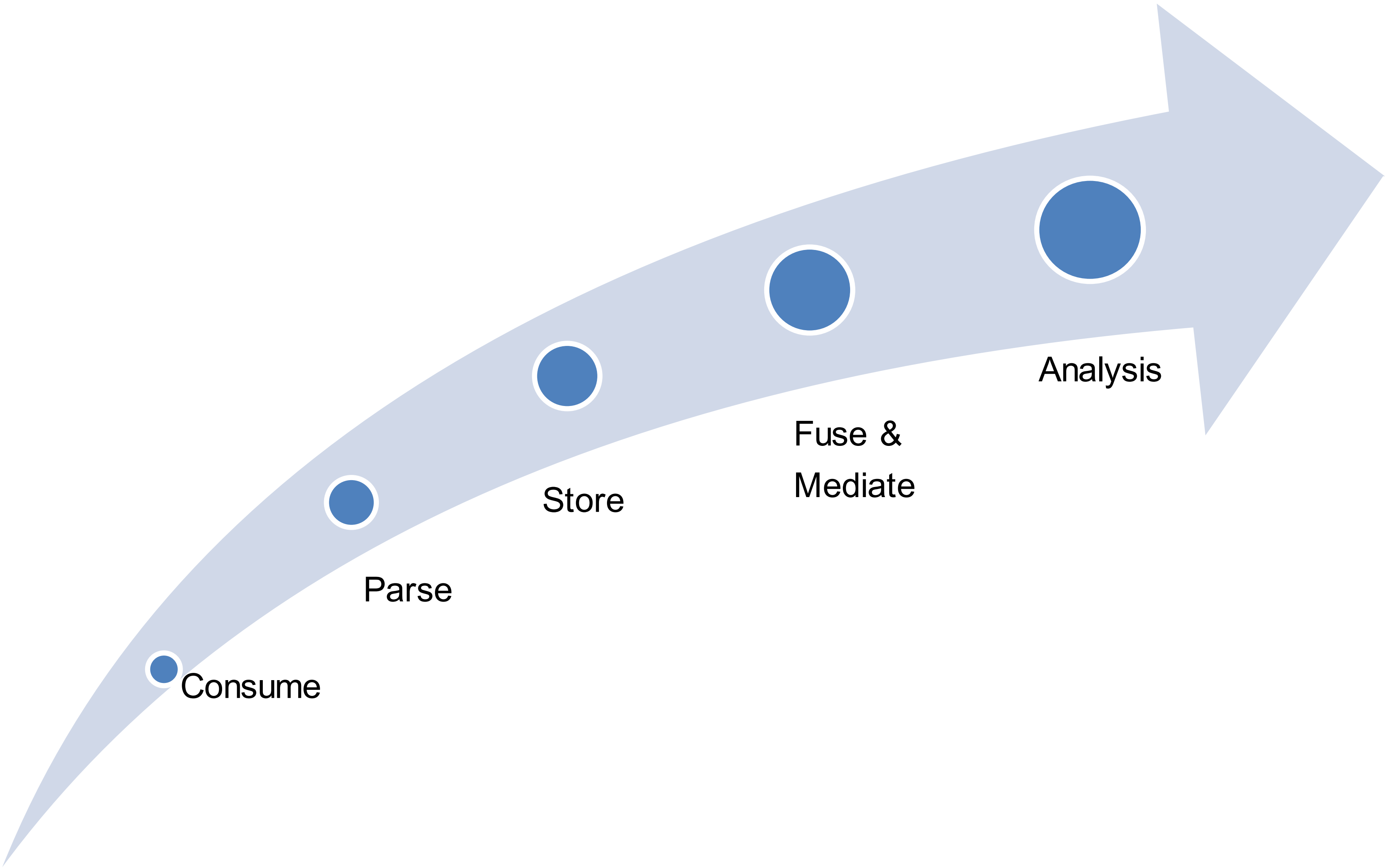
- Approach
  - Pull the data on the flight and all related flights from the GUFI tables
  - Find when the first mismatch happened
  - Create a file with the GUFI corrected for every entry
  - Run file through unit test tool to identify problem

Use Case: Traceability / Debugging
Flight Matching Issues

- Use Excel mapping tool to quickly identify that ASDE-X data for arrival flight incorrectly matched to departure flight with the same callsign
  - JIA5485 from GSO to CLT
  - JIA5485 from CLT to TYS
- Determined root cause was that ATD-2 had received incorrect gate IN time message
  - Matching service marked flight as having arrived
  - ASDE-X matching logic would not match to a flight that had already arrived at the gate
- Updated logic to better handle case with incorrect gate IN time message


GUFI
🟧 JIA5485.CLT.TYS.170721.0915.0915.AIRLINE
🟪 JIA5485.GSO.CLT.170725.0032.0061.AIRLINE

Planned Roadmap



Consume

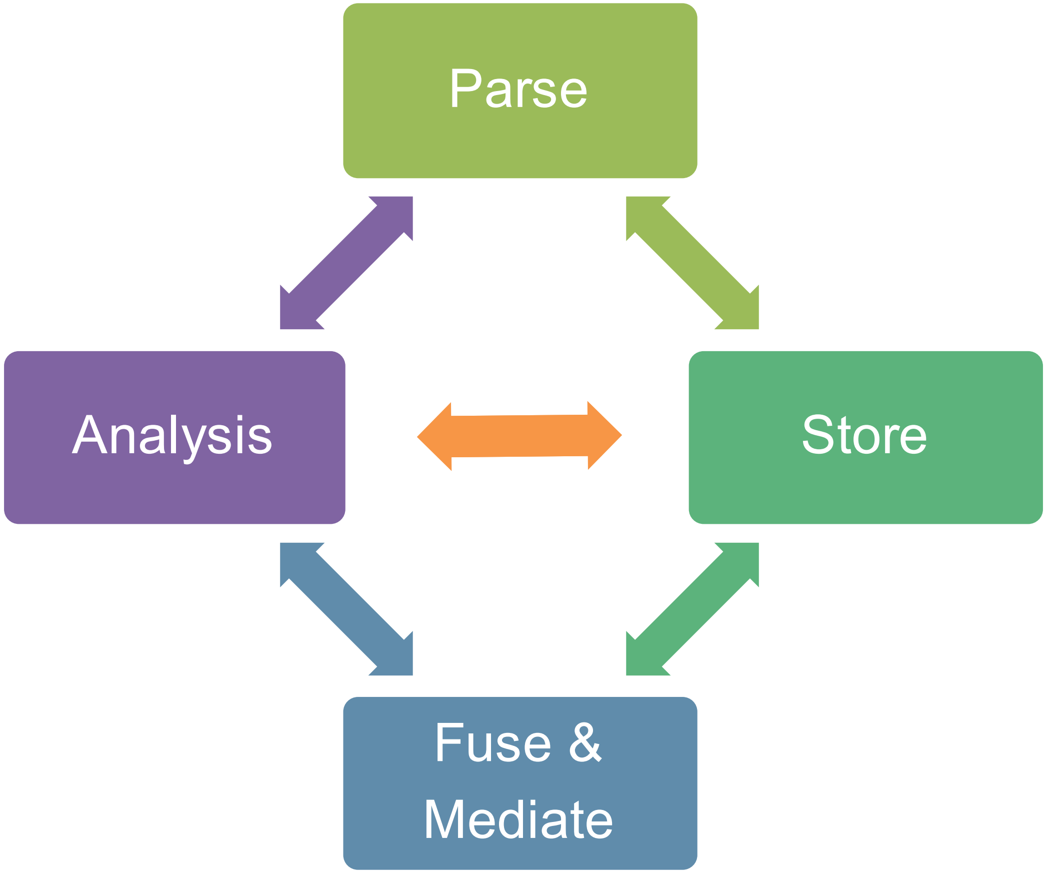
Parse

Store

Fuse & Mediate

Analysis

Actual Roadmap



Questions

Airspace Technology Demonstration 2 (ATD-2)

Turning SWIM Data into Consistent Reports for Analysts and Users

May 22, 2019

Background

- ATD-2 systems **ingest** huge amounts of SWIM data
- They also **output** huge amounts of data, recording every aspect of the operation

- *This output data is very valuable, but is too verbose to be used effectively for some analysis use cases*

- To address this challenge, we have developed a variety of standardized reports to serve analyst and user needs

Analyst challenges in using data

This output data presents several problems:

1. *Scale*: this is big data for most analysts, and Postgres query engine not forgiving for inexpert query design, particularly when trying to conduct longitudinal analysis
2. *Complexity*: DB design may seem complex to some analysts
3. *"Noise"*: human inputs, complexities of data mediation, order of processing messages, changes from earlier versions of ATD-2 software, etc.
4. *Business rules*: so many conventions for measurement

Create standardized reports to support analyst and user needs

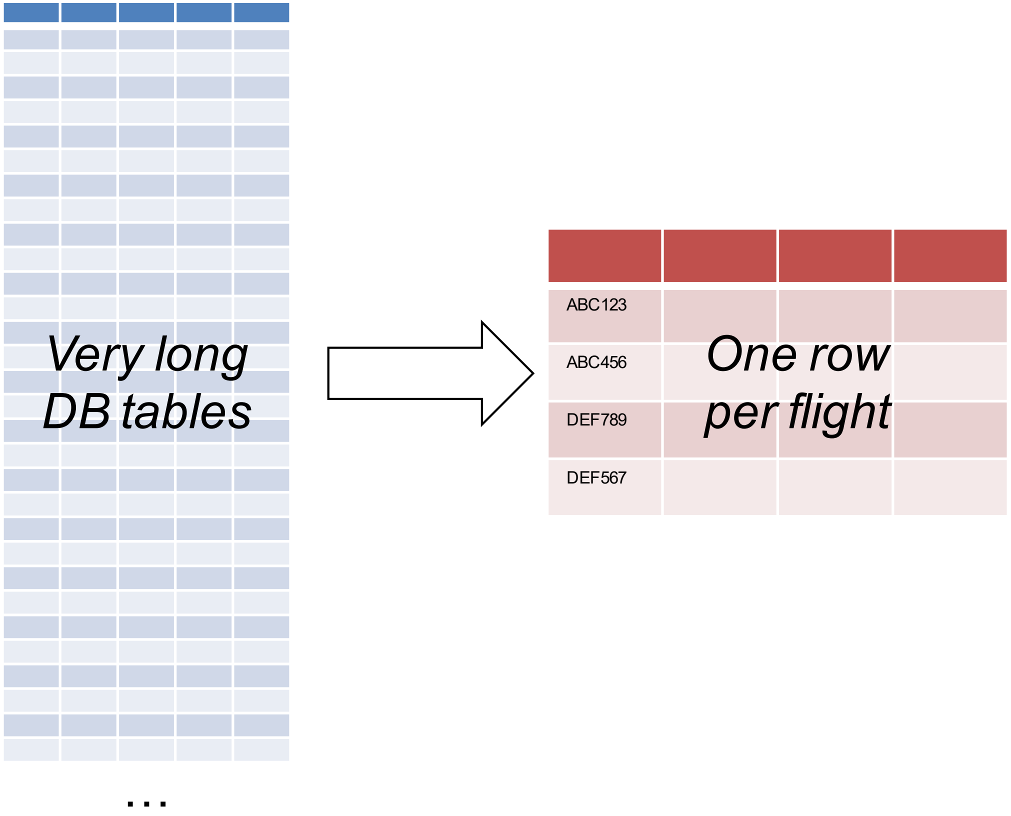
Sample of reporting artifacts

All reports possible because of consolidated view provided by fuser

- *flightSummary report*
  - Tabular report generated each day, one row per flight, *many* computed metrics
- *APREQ compliance report*
  - Subset of flightSummary, covering APREQ negotiation and compliance pushed to users each morning
- *Post-Metering report*
  - Subset of flightSummary, covering metering performance immediately after each bank at Charlotte
- *Daily Data Digest*
  - Summary of prior day's operation pushed to users and researchers each morning

flightSummary report introduction

- Fully compatible for all ATD-2 airports
- Report generated on data warehouses each morning for prior “day” (0400-0400 local), requiring ~15 minutes
  - Application written in Python, runs ~50 SQL queries, joins results, adds additional columns leveraging data between queries
  - Approach is generic: could be implemented in other languages, or in pure SQL



High-level sections of flightSummary

- “Basic” data
- Banks
- Clearances
- Flight “states”
- Surface metering
- Time/resource predictions at events
- EOBT, LTIME & associated accuracy
- EDCT, MIT, GS & Fix closures
- APREQ
- Undelayed/actual/excess taxi times
- AOBT by source
- Gate conflicts
- Airport configuration
- Predicted in times for departures
- AEFS
- First surveillance
- On-time performance

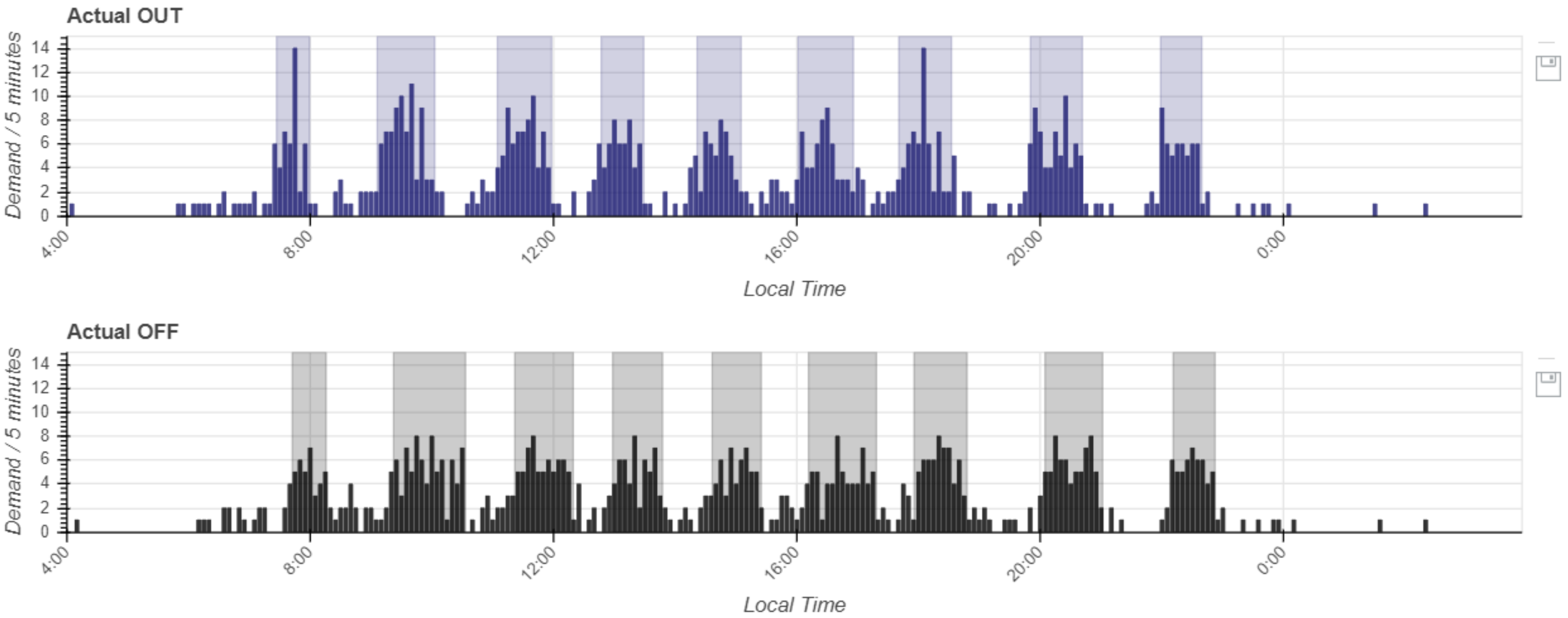
“Basic” descriptive data

From final values for each flight, report:

- ACID, GUFI
- Category, origin/destination
- Aircraft identifying info (type, wake, engine class, etc.)
- SOBT, SIBT
- AOBT, AMAT, ATOT, ALDT, AIBT (+ queue entry time)
- Actual terminal/ramp/gate/spot/runway/fix
- Cancellation indicator/time
- Final position
- Final route, assigned altitude
- Mainline/regional indicator
- Last system providing data, last timestamp of data received
- Long on board, priority status, runway opnec indicators
- IOBT, Final PTIME

Bank data

- Use clustering algorithm to infer banks from schedule and actual operations
  - Calculated for: scheduled in/out, actual on/off, actual out
  - Density-based clustering used, so some flights fall into no bank, representing lulls in traffic
- Also report operator-defined bank numbers, when available



Clearances

- RTC records all ramp controller actions, report gets last time each clearance issued
  - Gate pushback hold, gate pushback approved, proceed to spot, hold, return to gate, “not set”, cleared to gate
- Indicator for “true” return to gate status
  - Often observe controllers quickly undo clearance, pushing flight into unset state
  - Logic requires >5 minutes between clearance going return to gate or unset, and next good clearance, to count
- Indicator for pushback approved clearance being undone
- Last clearance type issued
- Infer pushback duration by difference between pushback approved and proceed to spot
  - Only captures flights cleared using RTC, as surveillance does not give reliable pushback duration

Flight states

- ATD-2 internal model maintains state of flight, based on available data and rules
  - Scheduled, pushback, ramp taxi out, taxi out, in queue, off, in terminal airport, en route
  - On final, taxi in, ramp taxi in, in gate
  - Return to gate, cancelled, suspended, unknown
- Query gets first time flight enters each state
- Report final state reached (helps with finding “stuck” flights)

Surface metering analysis

Developed suite of metrics around surface metering:

- Some values computed here apply to all flights, while others are specific to metered flights
- Infer flight ready time: capture clearance sequence, observation of surveillance, account for return to gate:
  - Report predictions at ready: controlled times, UOBT, UTOT, TOBT, TMAT, TTOT
- Infer metering "status"
- Standardized TOBT/TMAT compliance: using metering status and standard windows (TOBT +/- 2 mins, TMAT +/- 5 mins)
- Gate holds: advised and actual
- Held beyond SOBT or LTIME
- Fuel/emissions savings associated with actual gate hold
- Bulk of this data distributed after each bank for common situational awareness as the *Post-Metering Report*

Resource predictions at events

- For departures, immediately before:
  - Pushback, spot crossing, queue entry, off, fix crossing
- Predict:
  - Gate, spot, runway, fix (for all "future" resources)
- For arrivals, immediately before:
  - Fix crossing, landing, spot crossing, in
- Predict:
  - Fix, runway, spot, gate (for all "future" resources)

- Include data source for each resource prediction, e.g., STBO prediction, TBFM data

Time values & predictions at events

- At same events that resource predictions are sampled, get many times (set tailored to event):
- *Departures:*
  - At pushback: suite of gate (UOBT, LTIME, etc.), spot, runway (controlled, undelayed, etc.), fix times (targeted, undelayed, etc.)
  - At spot crossing: suite of spot, runway, fix times
  - At queue entry: suite of runway, fix times
  - At takeoff: suite of runway, fix times
  - At fix crossing: suite of fix times
- *Arrivals:*
  - Undelayed times for all future resources

EOBT / LTIME data

For each of EOBT and LTIME, report…

- Value at pilot ready time, at pushback clearance
- Final value received
- Difference versus ready time, pushback clearance, AOBT (using value in effect at that instant)
- EOBT at prescheduling
- Time first/last value received
- Number of times value updated
- Accuracy versus ready and AOBT at 0, 5, 10, 15, 20, 30 minutes prior to event

TMI impacts

EDCT:

- Values at pilot ready time, final
- When first/last EDCT received
- Number of updates
- Actual & truncated compliance

MIT & Fix closures:

- First/last time received
- Count of distinct restrictions

Ground stop:

- Indicator for data received

➢ *Area of active development to improve metrics*

APREQ data

Significant undertaking to include everything…

- First/actual release type (original, IDAC, free), coordinating center, time requested (if known)
- First/last scheduled times, TBFM-assigned delay
- First/last times flight scheduled, flight states at those
- Point in flight lifecycle when scheduled (e.g., pre pushback)
- Number of times rescheduled
- Time & fuel savings from rescheduling
- Actual & truncated compliance
- Prescheduling indicator, EOBT at prescheduling
- Bulk of this data distributed each morning to support analyst and user needs, common situational awareness

Taxi times

Undelayed:

- Record prediction used in system for undelayed taxi times, immediately before:
  - Pushback → ramp taxi time
  - Departure spot crossing → AMA taxi time
  - Landing → AMA taxi time
  - Arrival spot crossing → ramp taxi time
- Filter out “bad” values, include logic to account for bugs in historical data

Actual:

- Actual AMA & ramp taxi times for arrivals and departures
- Report excess (difference between actual and undelayed) taxi times for each phase

Other data

AOBT by source:

- Get AOBTs from:
  - Controller inputs (gate pushback approved)
  - Airline (CLT does not currently use these in operation)
  - Surveillance (occasionally, although coverage quality is low near terminal buildings)
- Often capture multiple airline-provided AOBTs because of different automation systems

Gate conflicts:

- System models/predicts gate conflicts, so capture data for both arrivals and departures
  - Associated other flight
  - Value present at landing (for arrivals)
  - Start/end/duration of conflict period (as of landing time)

Other data

Airport configuration:

- At out, off, on, in events for flights, record:
  - flow: direction airport operating in (small set of values for subject airports)
  - scenario: summary of departure procedures in effect

Downstream times for departures

- For departures from subject airports, report in time as predicted by airline systems, sample at out and takeoff events
- Useful for analysts to model downstream A04/A14 performance impacts

On-time performance:

- Report indicators for flights meeting D0/D15/A0/A14 milestones
- Use actual times truncated to minutes to match logic employed by DOT (as airline-provided times typically truncated)

Other data

AEFS actions:

- Cleared for takeoff
- Line-up and wait
- Enter runway
- Taxi clearance

***First surveillance data*:**

- Time of first surveillance data
- System providing first surveillance
- Flight state at first surveillance
  - Useful for understanding if flights pop into system before expected

Wrap-up

- These reports widely used within project as starting point for analysis, saving considerable redundant work
- Versions shared with project partners regularly for their analysis and feedback
- Development of these reports highly collaborative, adding new features regularly
- Approach is generic, but can be adapted as appropriate
- Infinitely simpler by starting with fuser data

- *This is current ATD-2 approach, but for future work, we believe that maintaining a common 360° view of each flight is extremely valuable*.

Airspace Technology Demonstration 2 (ATD-2)

Accuracy Comparison of Various Landing Time Prediction Sources

May 22, 2019

Objective and background

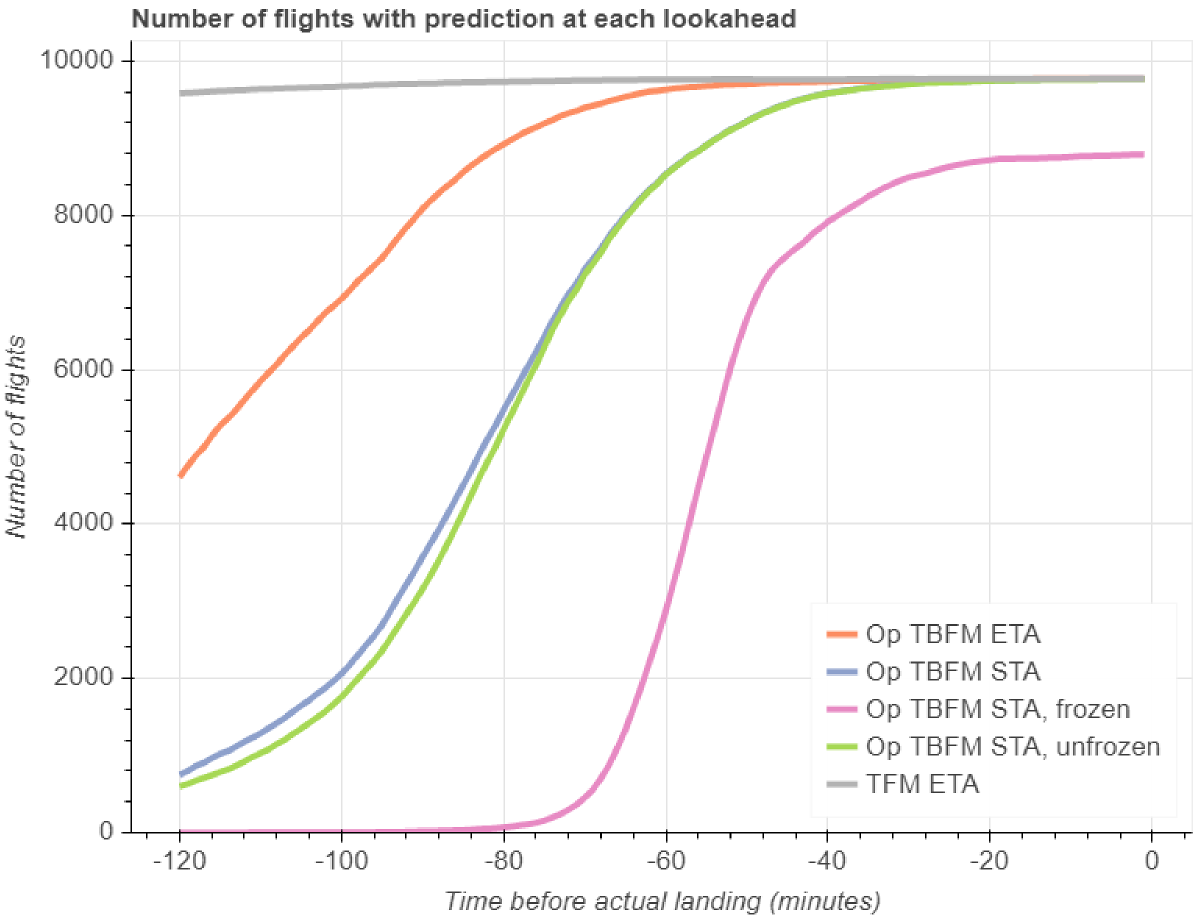
Quantify accuracy of different sources of landing (on) time predictions as actual arrival event approaches

- Want to make design decisions for fuser mediation rules informed by data about actual accuracy of various potential prediction sources
- Accuracy defined as difference between actual and prediction, particularly interested in how this evolves as actual event approaches
- Other sources could easily be included in this framework, e.g., operator-generated predictions

Data and methodology

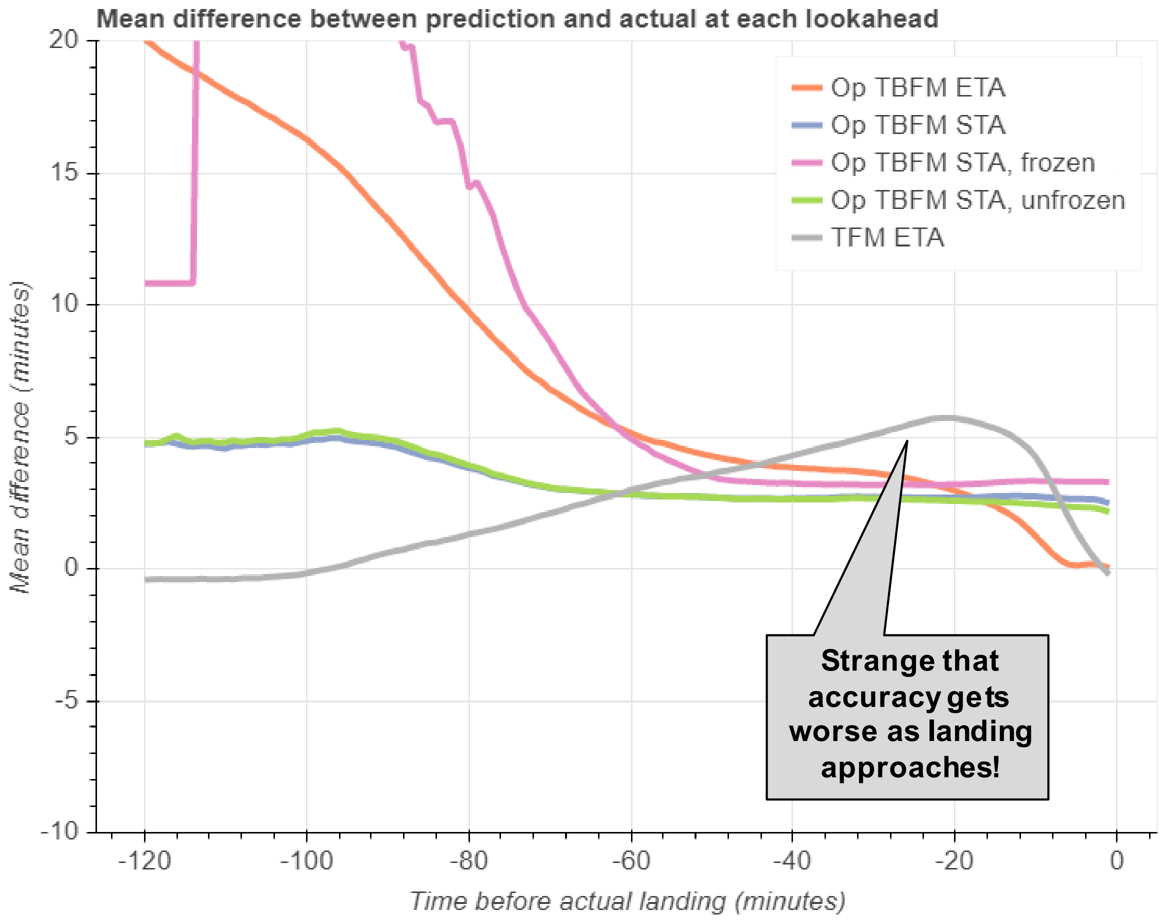
- Two weeks of arrivals to Charlotte from March 2019
- Comparing:
  - TBFM ETA
  - TBFM STA
  - TBFM STA (only when frozen)
  - TBFM STA (before frozen)
  - TFM ETA
- Measure error as actual landing time – prediction
- Sample every minute for every flight, then average in a variety of ways

How many flights have predictions?



- Consistent with expectations – TFM available well ahead of time, then TBFM ETA, then STAs begin appearing and are frozen

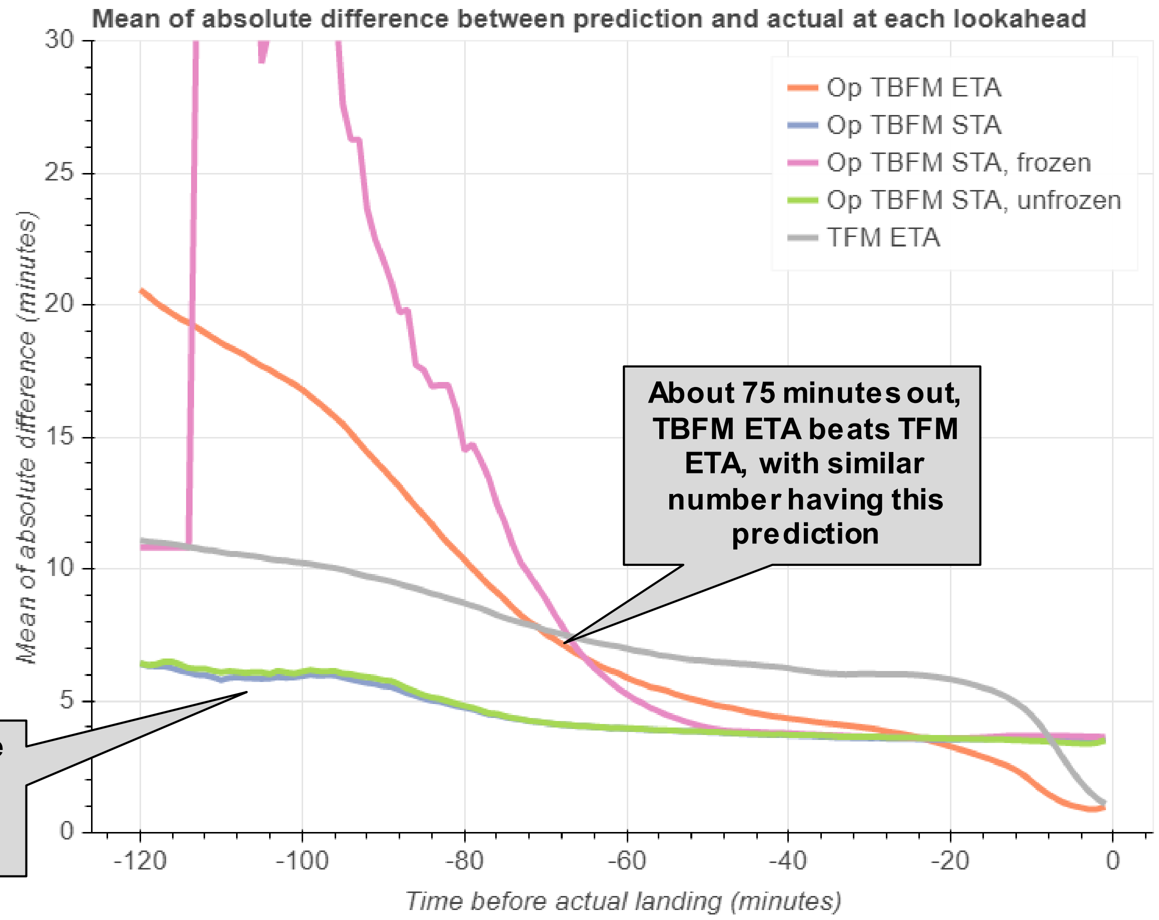
What are the mean errors?



Strange that accuracy gets worse as landing approaches!

- Mean is a potentially troublesome measure because positive and negative errors may cancel each other out

What are the *MAD* errors?

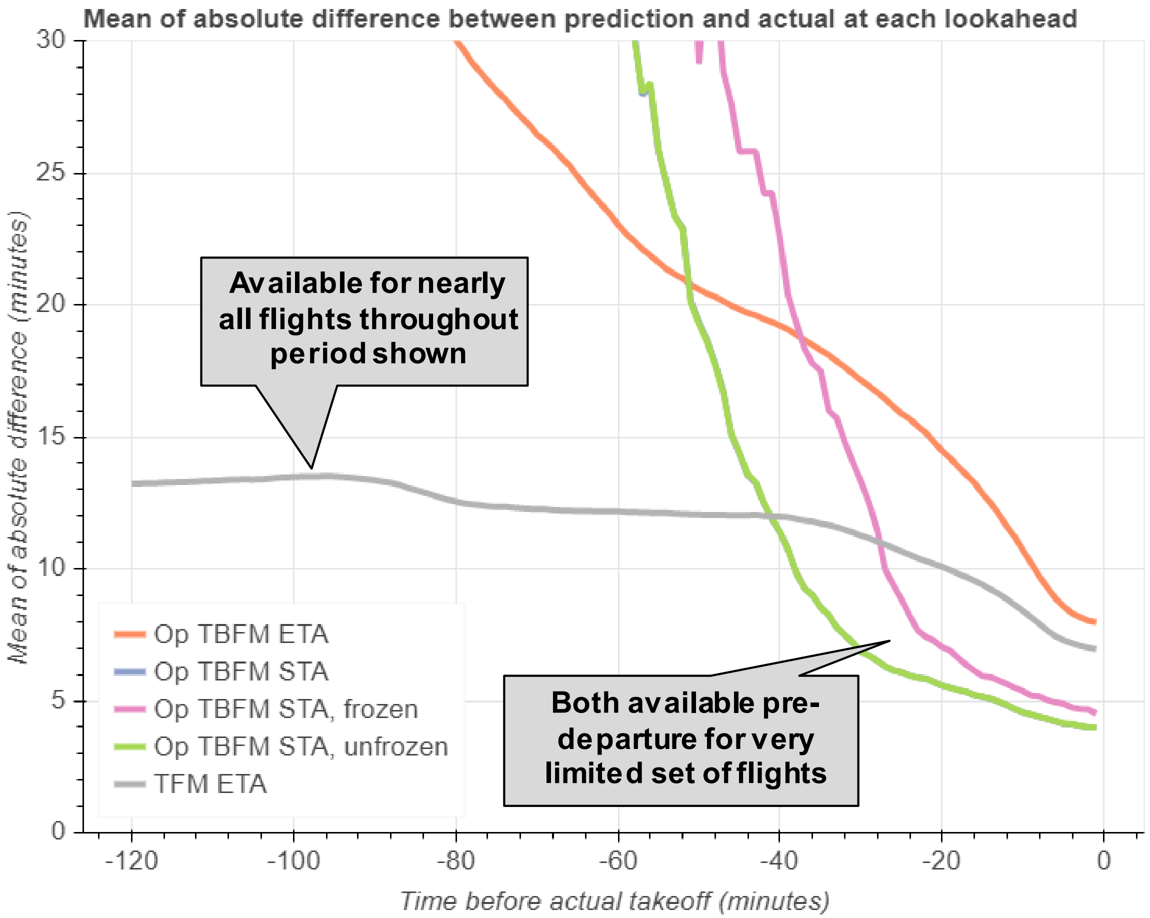


About 75 minutes out, TBFM ETA beats TFM ETA, with similar number having this prediction

Recall that there are few flights with this prediction

- Plot shows **mean of absolute difference** (error), weighting positive and negative errors equally, but showing best measure of "average" error

Are predictions worse before takeoff?

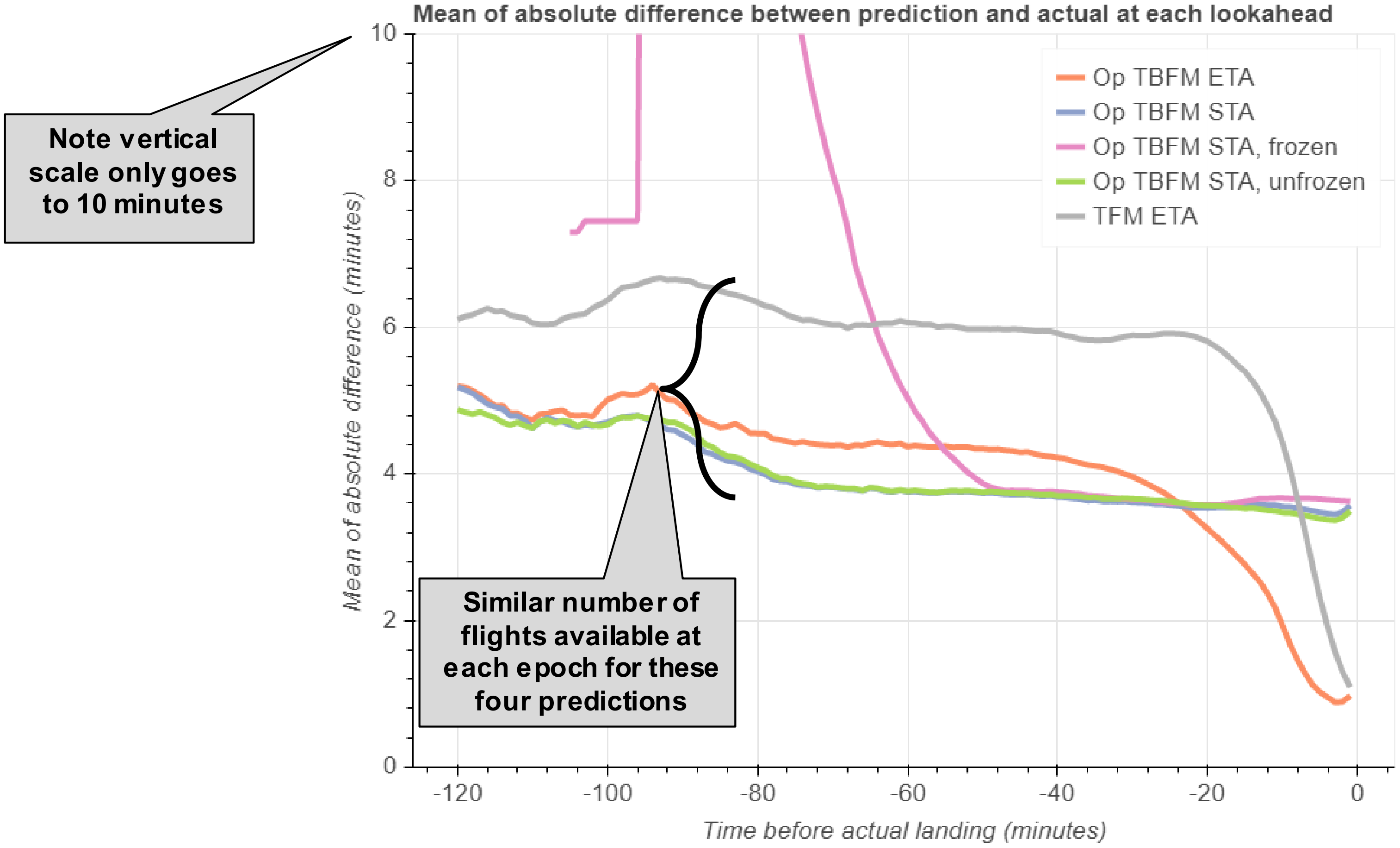


Available for nearly all flights throughout period shown

Both available pre-departure for very limited set of flights

- Plot shows MAD leading up to takeoff time for same set of flights. Only TFM ETA widely available, but TBFM ETA becomes available 60-80 minutes pre-departure
- Conclusion: predictions somewhat poor pre-departure

Are predictions better after takeoff?



Note vertical scale only goes to 10 minutes

Similar number of flights available at each epoch for these four predictions

- Plot shows MAD leading up to landing time, but only includes flights that have already departed
- Post-departure, predictions are much better than pre
- Seems clear that TBFM provides best estimates at most lookaheads

Wrap-up

- Prediction accuracy generally improves as landing time approaches, as expected
- Not all errors converge to zero
- Demonstration of feasibility of comparing landing time prediction accuracy of various data sources
  - *This work could be replicated with a big pile of data captured directly from SWIM feeds*
  - ***But, this work is significantly simpler when using data that has passed through the fuser / ATD-2 system***

Airspace Technology Demonstration 2 (ATD-2)

Analysis of APREQ Flights at CLT

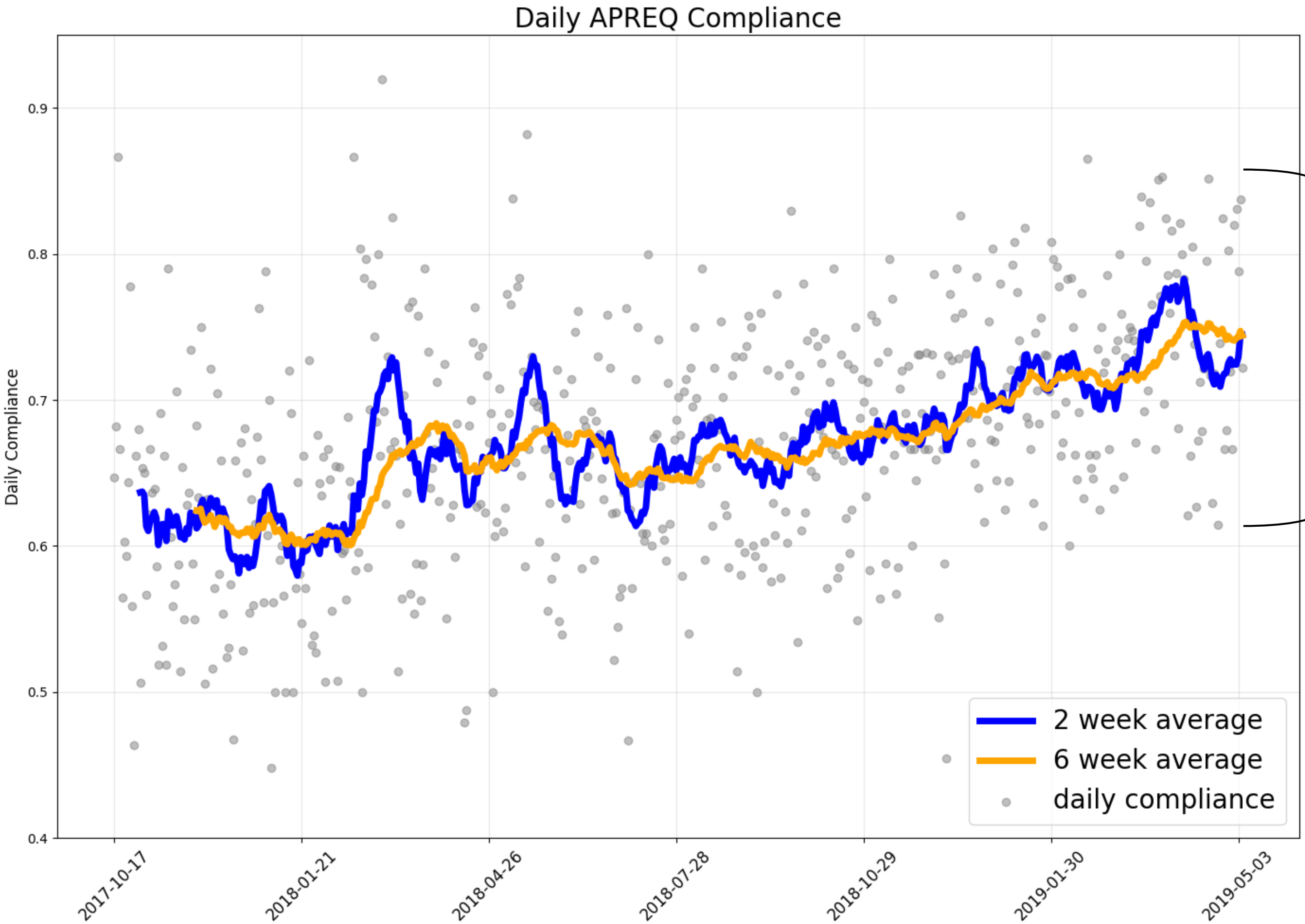
May 22, 2019

Objective

Quantify impact of IADS Phase 1 & 2 capabilities on APREQ flights at CLT with respect to:

- Compliance to the Controlled Take Off Time (CTOT)
- Benefits for APREQ flights that use IDAC to renegotiate for an *earlier* CTOT
- Benefits of pre-scheduling APREQ flights using the Earliest Off Block Time (EOBT)
- Relationship between EOBT compliance and rescheduling CTOT

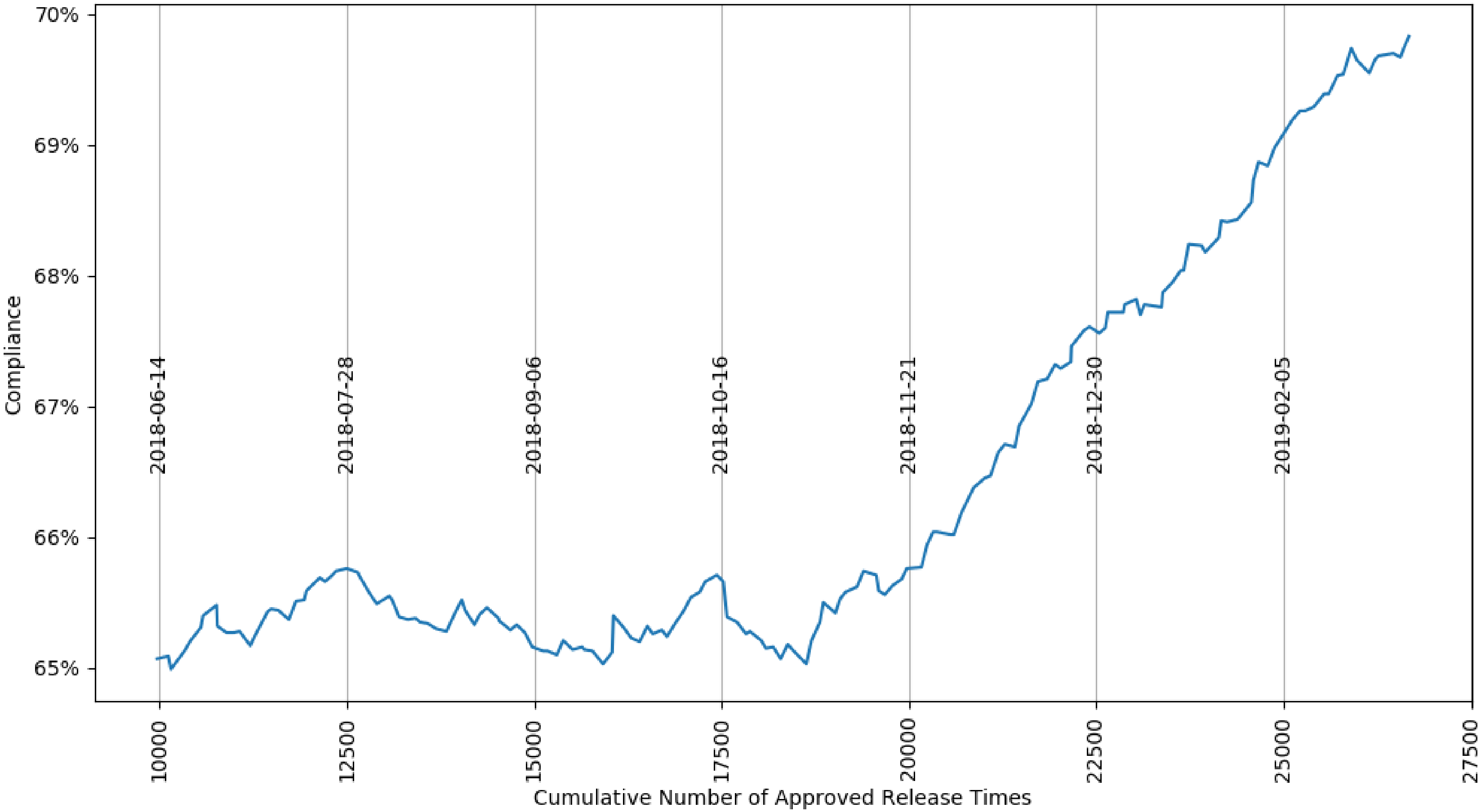
CLT APREQ Daily Compliance (Compliance Improvement Since ATD-2 Start)



In addition to overall improved compliance into TBM systems, the predictability is ***increasing***

Steady increase of APREQ compliance over the life of the project.
Reduced variation in compliance leading to improved predictability.

APREQ Compliance 10K Rolling Window



The most substantial APREQ compliance improvements started with Phase 2 capability (AEFS integration, ZTL IDAC, pre-scheduling and scheduler updates).

IADS Phase 1 & 2 Benefit Mechanisms

1. **Collaborative surface metering**
   - Reduced engine run time
   - Reduced fuel consumption and emissions

2. **Overhead stream operational integration**

   a. Scheduling controlled flights at the gate
   - Reduced engine run time
   - Reduced fuel consumption and emissions

   b. APREQ renegotiating for an earlier slot
   - Reduced total delay
   - Passenger value of time and crew costs
   - Reduced engine run time
   - Reduced fuel consumption and emissions

Benefits (1) and (2a) achieved through tactical gate holds

Benefit (2b) achieved through APREQ renegotiation process described below

Step 1: APREQ flight has a release time but is capable of taking off earlier

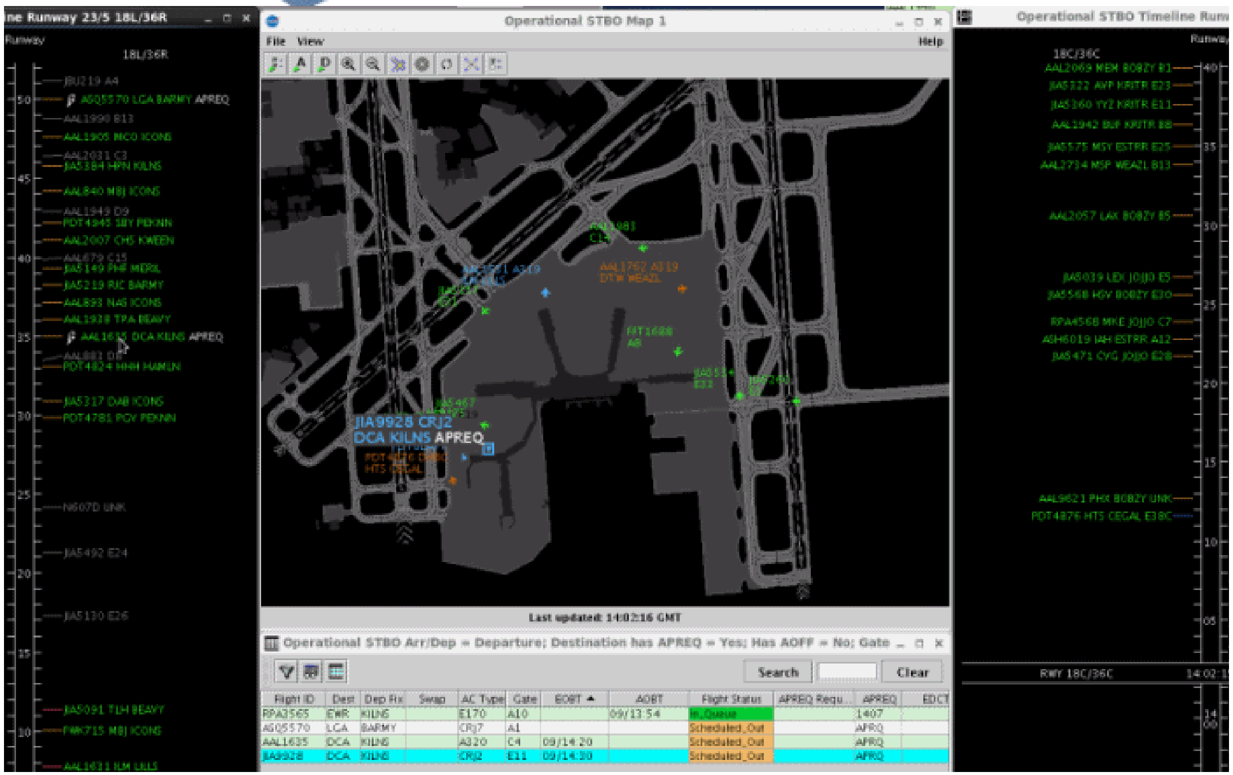
Step 2: FAA TMC uses the IDAC green space / red space to identify and request an *earlier* slot in the overhead stream

Step 3: Aircraft receives *earlier* release time and the difference between the release times is the reduction in delay

ATD2 Integrated Arrival/Departure/Surface

Benefits for APREQ flights using IDAC to renegotiate for *earlier* CTOT

270.7 hours of delay saved by electronically renegotiating a better overhead stream time for 2,071 flights.

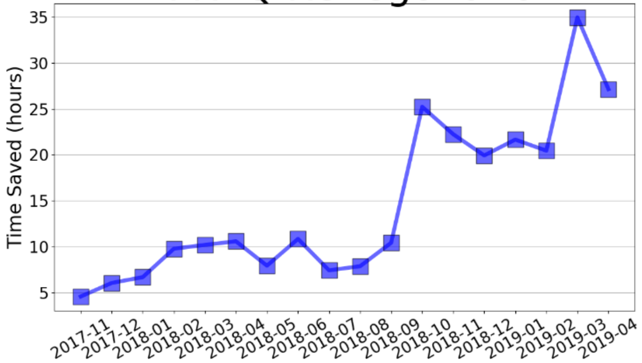


Last updated: 14:02:16 GMT

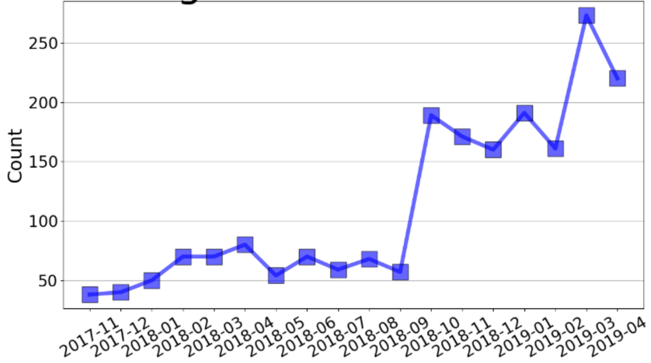
Operational STBO Arr/Dep = Departure; Destination has APREQ = Yes; Has AOFF = No; Gate

Flight ID	Dest	Dep Fix	Swap	AC Type	Gate	EOBT ▲	AOBT	Flight Status	APREQ Requ...	APREQ	EDCT
RPA3565	EWR	KILNS		E170	A10		09/13:54	In_Queue		1407	
ASQ5570	LGA	BARMY		CRJ7	A1			Scheduled_Out		APRQ	
AAL1635	DCA	KILNS		A320	C4	09/14:20		Scheduled_Out		APRQ	
JIA9928	DCA	KILNS		CRJ2	E11	09/14:30		Scheduled_Out		APRQ	

Time Saved by IDAC-related APREQ Renegotiation

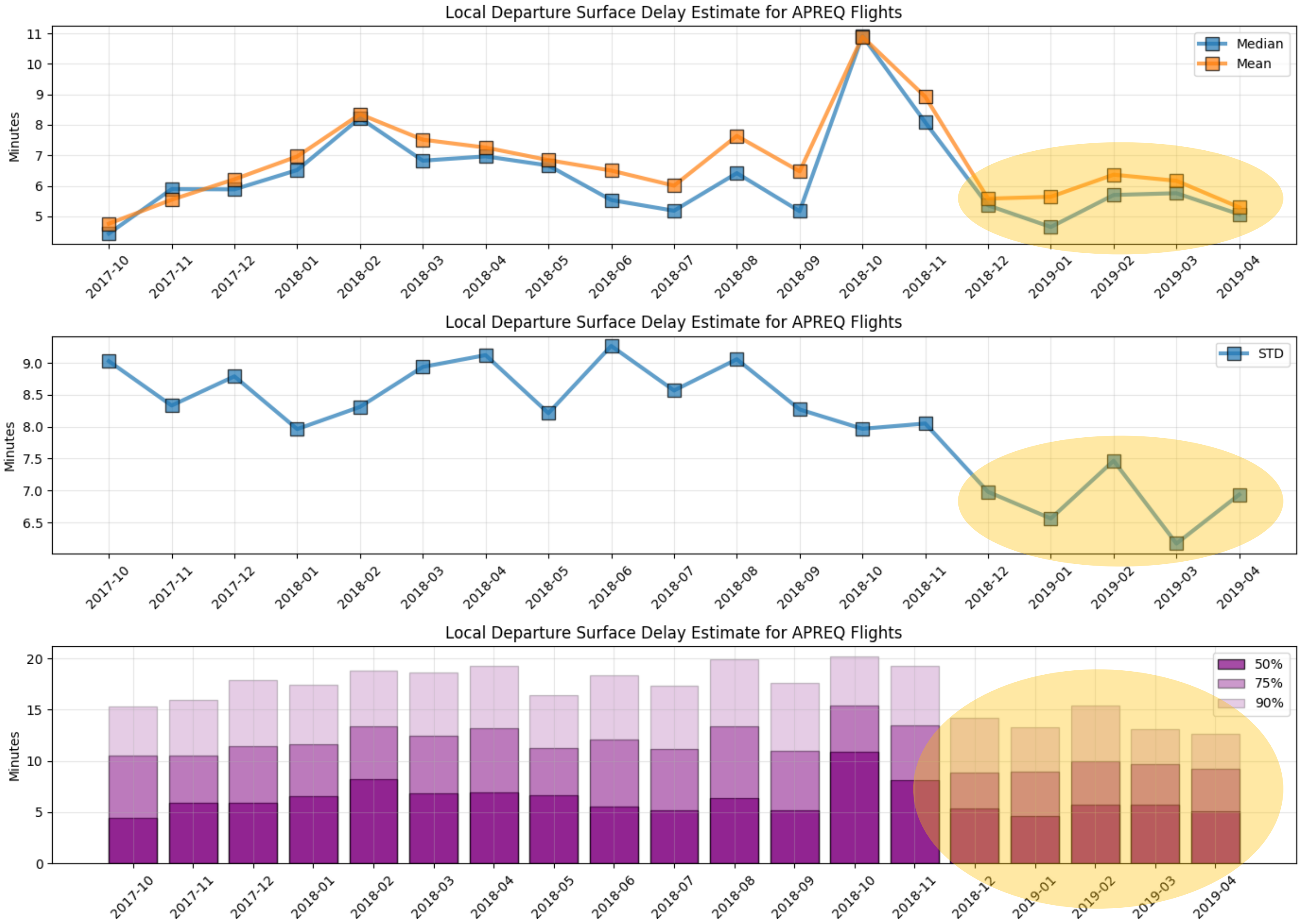


Count of Departures that Renegotiate for Earlier Time



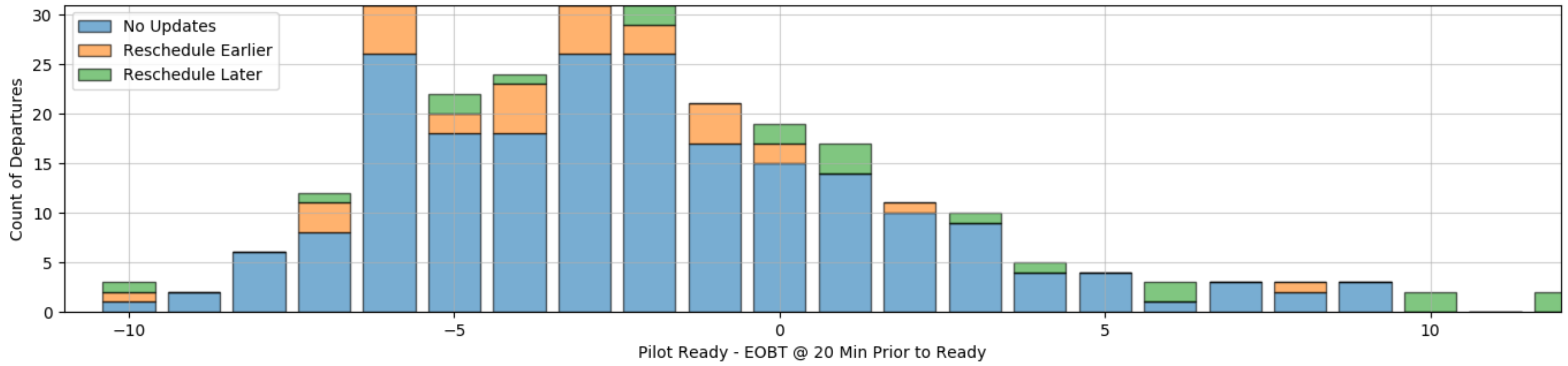
- The benefits described here are associated with better use of existing capacity in the overhead stream, and technology to reduce surface delay.
- These benefits are in addition to (distinct from) surface metering savings.

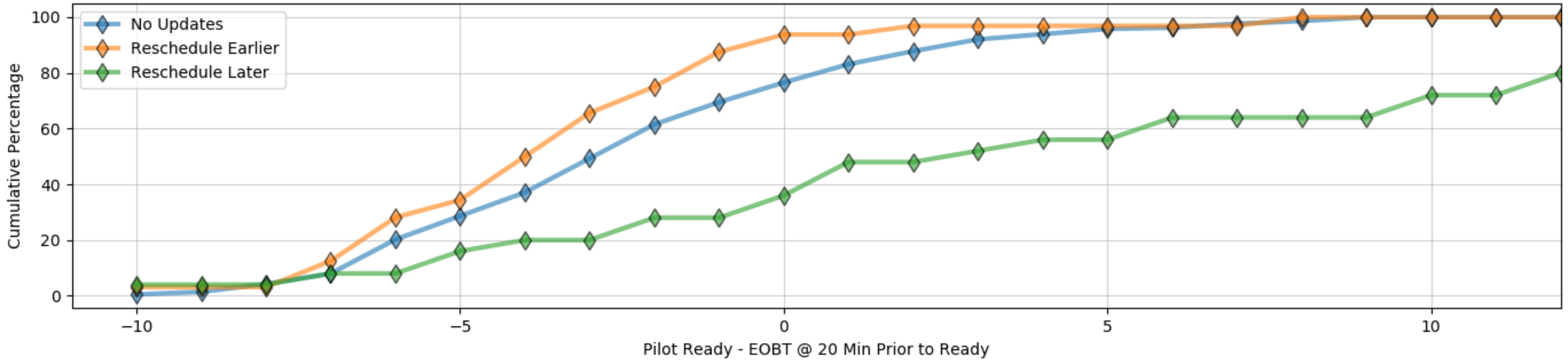
APREQ Delay For Pre-Scheduled Flights into KATL Have Been Reduced and are More Predictable For the Last Five Months

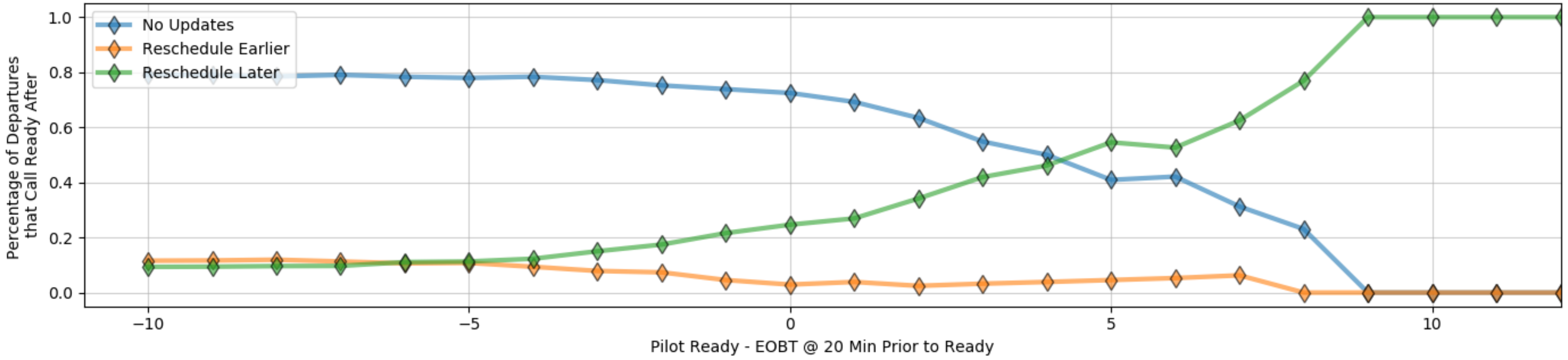


Substantial Improvements in predictability of delay for the last 5 months

EOBT Compliance / CTOT Reschedule for Pre-Scheduled Flights into KATL







Wrap-up

- Compliance to the CTOT has improved throughout the lifecycle of ATD-2 with biggest improvements following the introduction of Phase 2 capabilities

- Rescheduling APREQ flights using IDAC has reduced 270.7 hours of delay at CLT

- Predictability of local surface delay for APREQ flights is substantially improved via pre-scheduling with the IADS system

- Pre-scheduled flights that reschedule for later times tend to call ready later with respect to EOBT

Airspace Technology Demonstration 2 (ATD-2)

Predictive Analytics for ATD-2

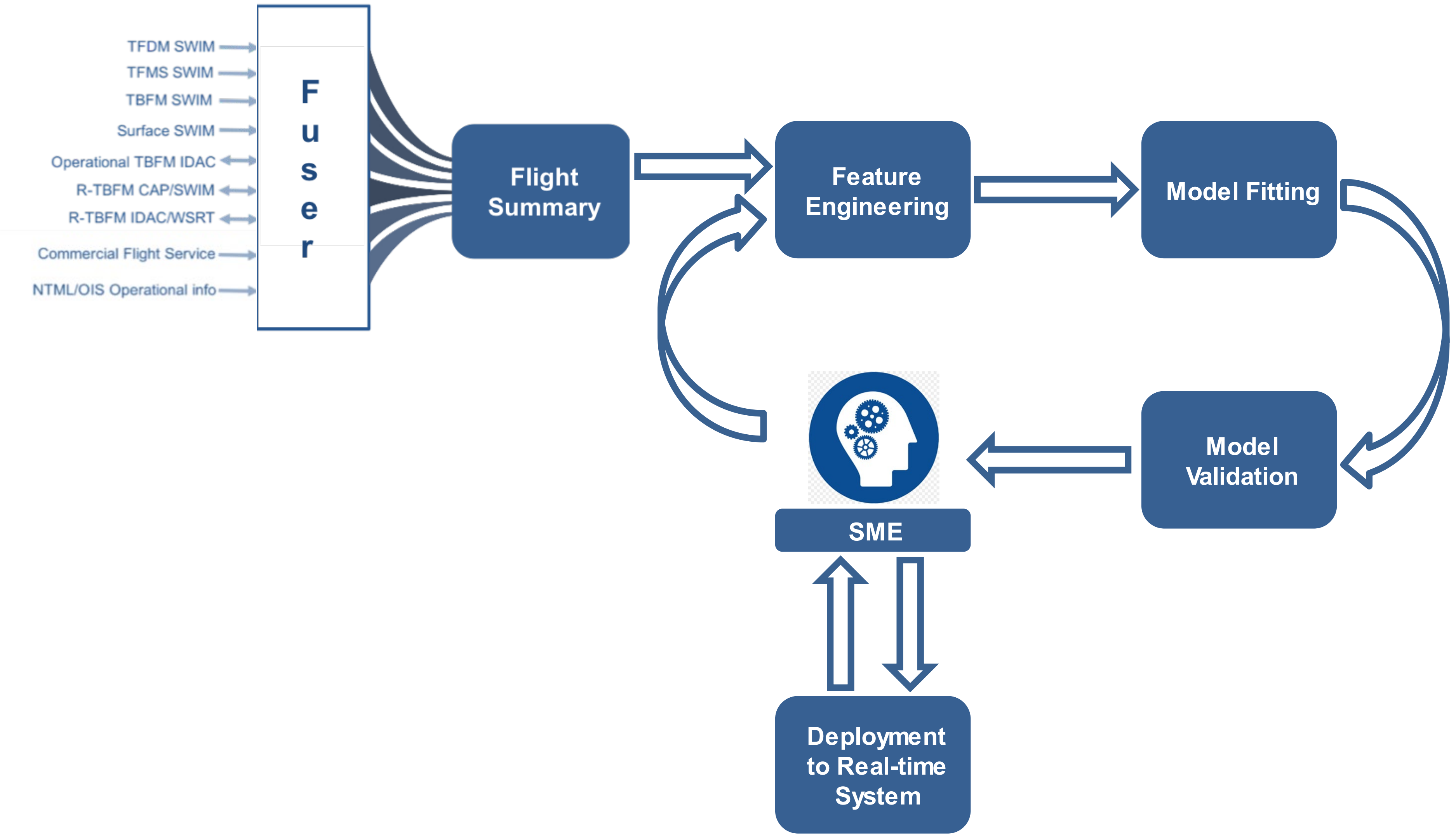
May 22, 2019

Objective

Leverage high quality data and predictive analytics to improve understanding and performance of IADS system

- Develop predictive analytics use cases that are relevant to FAA and operators
- Iterative process between data scientists and Subject Matter Experts (SME) to gain new insights
- Implementation in Python Scikit-learn allows for data scientists to focus on feature engineering and model validation
- Interested in data available in real-time system to fit models that have predictive and ultimately prescriptive capabilities

ATD-2 Predictive Analytics Workflow



Gate Conflict Use Case

Predicting gate conflicts can benefit both FAA and operators

- Providing ramp controllers with early notice of gate conflicts allows them to build a plan

- Providing FAA with early notice of gate conflicts supports the TMC in the decision whether or not to surface metering

- Understanding the different factors that cause gate conflicts could provide strategies to avoid them

Feature Engineering: Bank Level Metrics

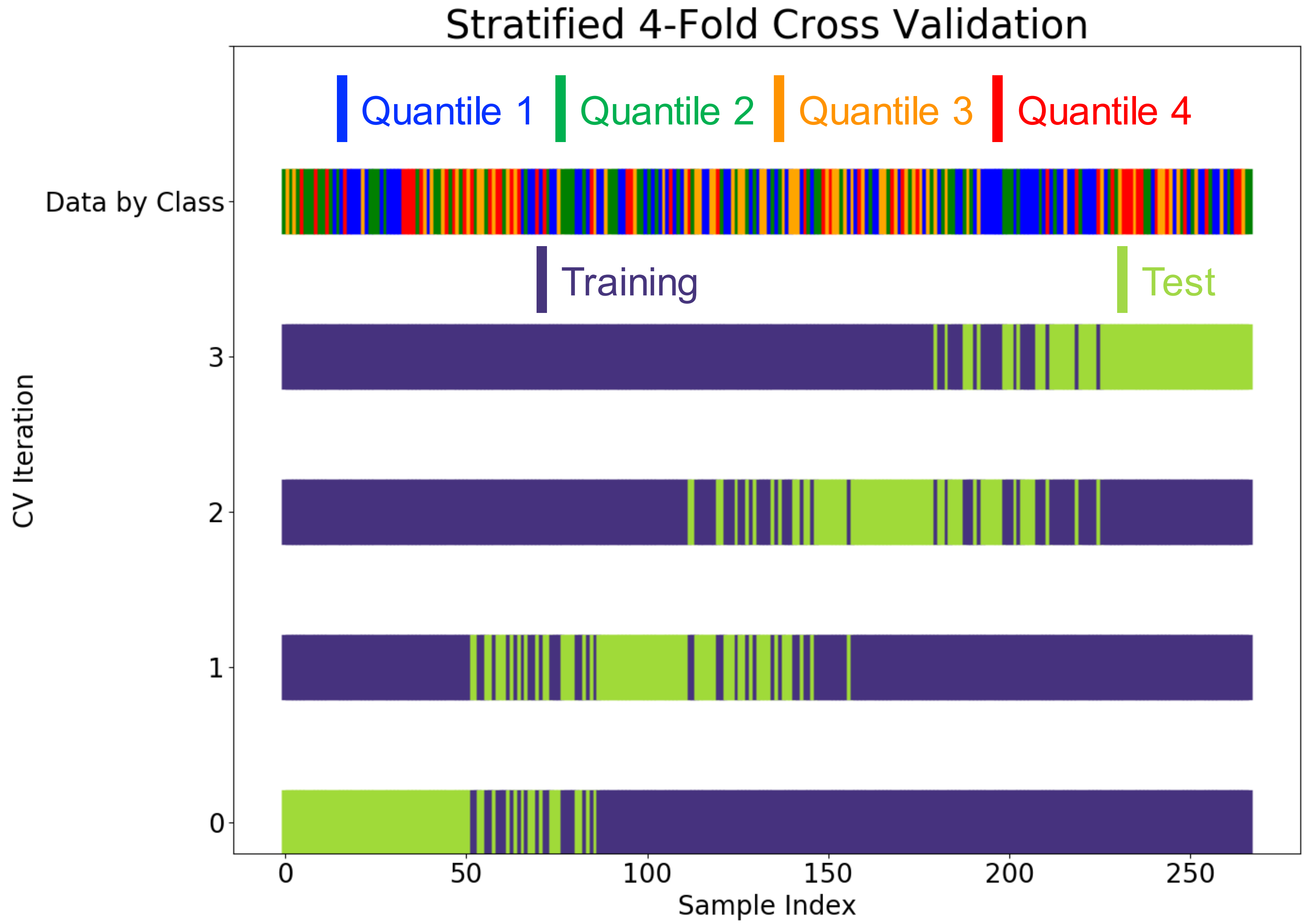
Bank Level Features

Regression Target

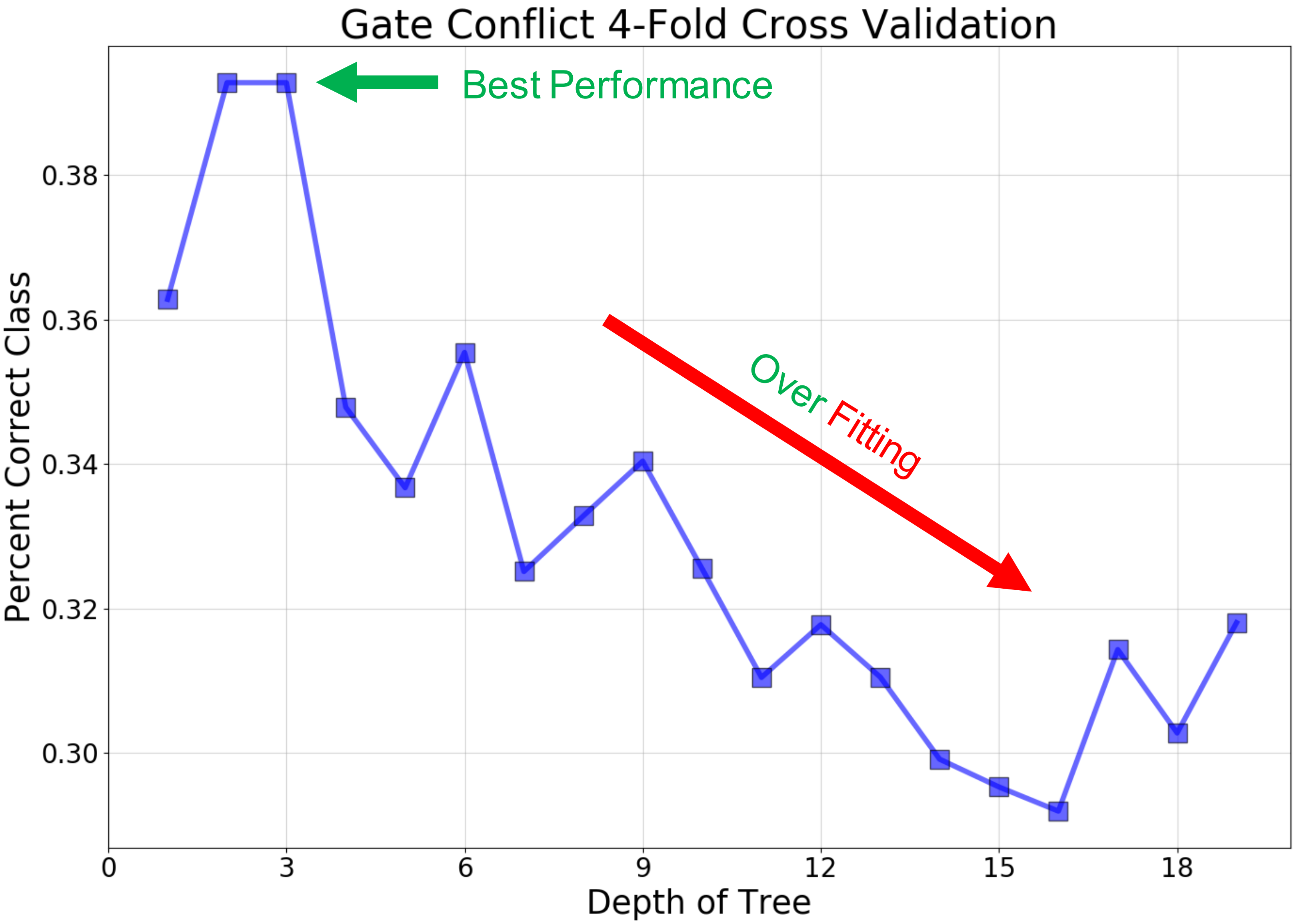
Classification Target

Date	Bank	Count Departure	Count Arrival	Difference in Dep and Arv Bank Start (bank_overlap)	Departure Gate Hold (total_actual_gate_hold)	. . .	Count Gate Conflict	Quantile Gate Conflict
2018-06-24	2	78	75	11.1	29.9		9	Quantile 3
2018-06-25	2	86	75	21.3	34.7		9	Quantile 3
2018-06-26	2	92	85	19.5	24.0		4	Quantile 1
2018-06-27	2	96	88	24.5	51.5		6	Quantile 2
2018-06-29	2	98	86	8.9	39.2		5	Quantile 1
2018-06-30	2	79	73	13.0	66.7		6	Quantile 2

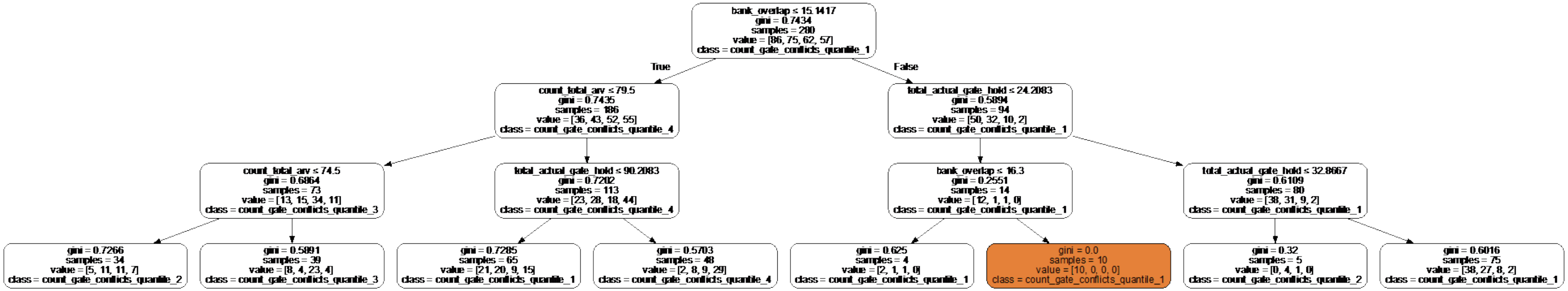
Gate Conflict: Stratified 4-Fold Cross Validation



Gate Conflict Decision Tree: Hyperparameter Tuning and Validation



Gate Conflict Decision Tree



bank_overlap ≤ 15.1417
gini = 0.7434
samples = 280
value = [86, 75, 62, 57]
class = count_gate_conflicts_quantile_1

True

False

count_total_arv ≤ 79.5
gini = 0.7435
samples = 186
value = [36, 43, 52, 55]
class = count_gate_conflicts_quantile_4

total_actual_gate_hold ≤ 24.2083
gini = 0.5894
samples = 94
value = [50, 32, 10, 2]
class = count_gate_conflicts_quantile_1

count_total_arv ≤ 74.5
gini = 0.6864
samples = 73
value = [13, 15, 34, 11]
class = count_gate_conflicts_quantile_3

total_actual_gate_hold ≤ 90.2083
gini = 0.7202
samples = 113
value = [23, 28, 18, 44]
class = count_gate_conflicts_quantile_4

bank_overlap ≤ 16.3
gini = 0.2551
samples = 14
value = [12, 1, 1, 0]
class = count_gate_conflicts_quantile_1

total_actual_gate_hold ≤ 32.8667
gini = 0.6109
samples = 80
value = [38, 31, 9, 2]
class = count_gate_conflicts_quantile_1

gini = 0.7266
samples = 34
value = [5, 11, 11, 7]
class = count_gate_conflicts_quantile_2

gini = 0.5891
samples = 39
value = [8, 4, 23, 4]
class = count_gate_conflicts_quantile_3

gini = 0.7285
samples = 65
value = [21, 20, 9, 15]
class = count_gate_conflicts_quantile_1

gini = 0.5703
samples = 48
value = [2, 8, 9, 29]
class = count_gate_conflicts_quantile_4

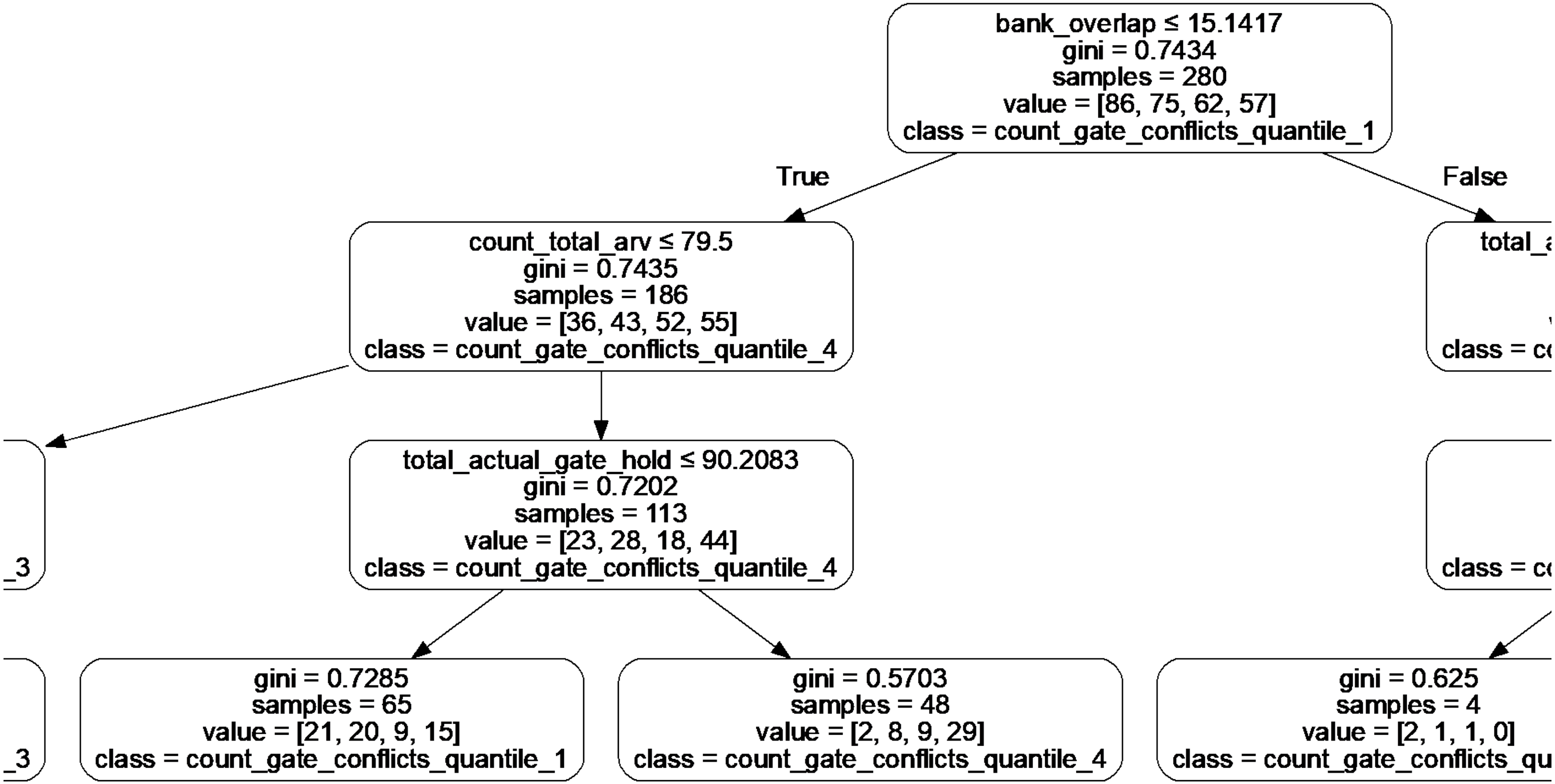
gini = 0.625
samples = 4
value = [2, 1, 1, 0]
class = count_gate_conflicts_quantile_1

gini = 0.0
samples = 10
value = [10, 0, 0, 0]
class = count_gate_conflicts_quantile_1

gini = 0.32
samples = 5
value = [0, 4, 1, 0]
class = count_gate_conflicts_quantile_2

gini = 0.6016
samples = 75
value = [38, 27, 8, 2]
class = count_gate_conflicts_quantile_1

Gate Conflict Decision Tree



Wrap-up

- High quality data is the foundation of predictive analytics
- Selecting and building features that best represent the problem is a critical step in the process
- Hyperparameter tuning in combination with cross validation to achieve the best performance
- Models are trained by data scientist and then evaluated by SME in iterative process
- Deployment to real-time system is necessary to achieve impact across the NAS

Airspace Technology Demonstration 2 (ATD-2)

TFDM Terminal Publication Service (TTP)

May 22, 2019

TFDM Terminal Publication (TTP)

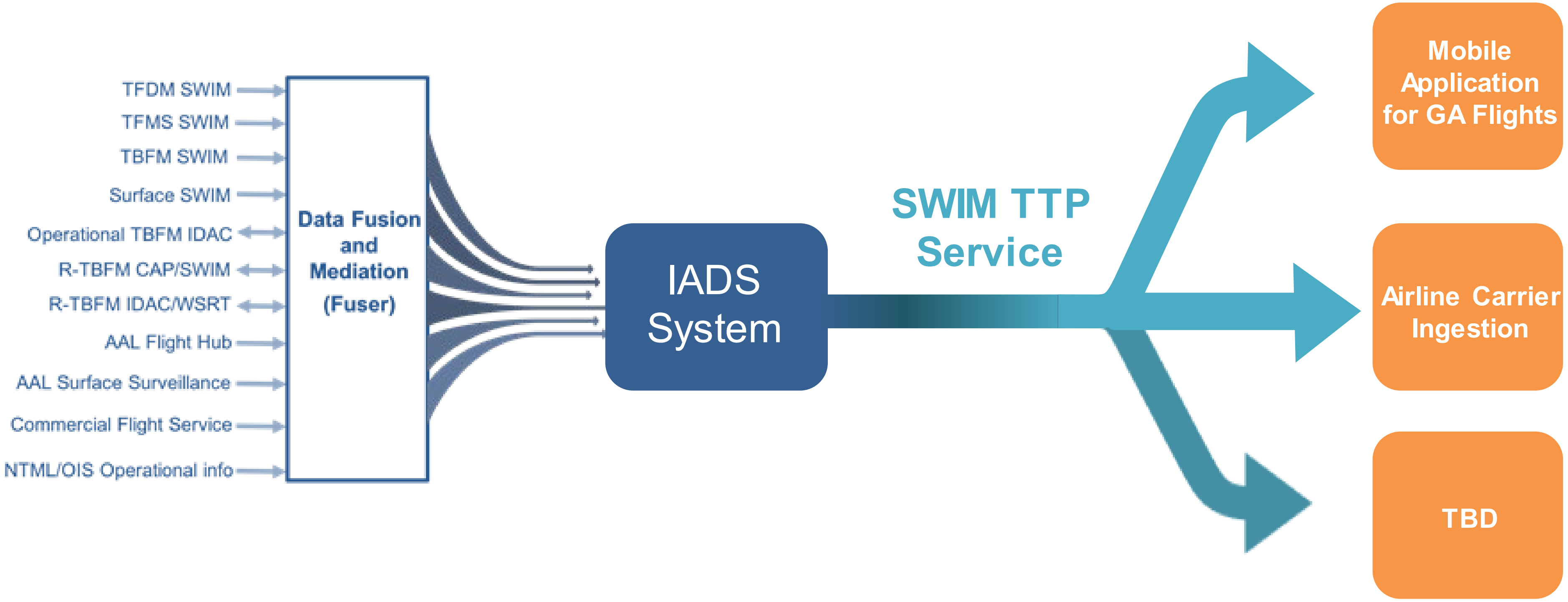
- TFDM data feed publishing Flight and Flow data to consumers
- Will provide data exchange between TFDM and NAS Systems and the National Airspace System (NAS) users (airlines, air carriers, air freight, military or general aviation/business aviation operators).
- Accessible via the National Airspace (NAS) Enterprise Messaging Service (NEMS).
- Uses the publish-subscribe (pub-sub) Message Exchange Pattern (MEP).
- XML data format, using FIXM standard for Flight Data
- Airport Information and Traffic Management Restrictions use a schema defined by the TFDM team

ATD-2 Implementation of TTP

- Registered as “NASA TTP” in NSRR
- Currently available via SWIM R&D Gateway
- Based on TFDM specifications
  - Currently no deviations from TFDM specifications
  - Does not include all information published by TFDM
- Publishing data for:
  - Charlotte Douglas International Airport
  - Dallas/Fort Worth Metroplex
- Planning support of NASA TTP for CLT until TFDM proper installed
- Goal - work invested in integrating with ATD-2 via TTP could be utilized with TFDM

IADS and Data Sharing

Applications that Leverage the TTP Prototype Feed



TTP Services

Service	Includes
Flight Data	Individual flight updates containing flight identifiers, targeted times, actual times, runway, parking gate, spot, departure fix (predicted, assigned, actual as appropriate), flight states, and more
Airport Information	Airport configurations, airport and runway rates, ramp closures, runway closures, taxiway closures
Traffic Management Restrictions	Call for Release programs departure MIT/MINIT restrictions, departure stop/ground stop programs. Along with list of impacted flights for each
Flight Delay	Airport and runway delay by arrival, departure, and total
Operational Metrics	Metrics on airport throughput and individual flight metrics
SMP	Data about Surface Metering Programs, affected flights, and metering parameters

TTP Services

Name	Event Driven	Full Update	Implemented in NASA TTP
Flight Data	Yes	Every 15 minutes	Yes
Airport Information	Yes	Every 15 minutes	Yes (subset)
Traffic Management Restrictions	Yes	Every 15 minutes	Yes (subset)
Flight Delay	Yes	Every 15 minutes	Yes (subset)
Operational Metrics	No	Every 1 minute	Yes (subset)
SMP	Yes	Every 15 minutes	Not currently

- We will continue to track and align with TFDM as much as possible
- Implementation details of specific messages can be found on NASA TTP NSRR

Why TTP?

- Share valuable data with other stake holders
- Automate data sharing avoiding manual inputs
- Data doesn’t exist in other feeds
- Doesn’t naturally fit into any existing feeds

Example Fields of Interest

- **Flight Data Fields**
  - APREQ Release Time
    - Approval Request Release Time / Call for Release Time received from TBFM
  - Departure Runway Predicted
    - The departure runway predicted by the STBO model
  - Departure Runway Actual
    - The departure runway the flight departed from
  - Arrival Runway Predicted
    - The arrival runway predicted by the STBO model
  - Arrival Runway Actual
    - The departure runway the flight departed from
  - Estimated Time of Departure
    - The time of departure predicted by the STBO model
  - TMI Identifiers
    - Contains a comma delimited list of TMI IDs, one per TMI associated with the flight

Example Fields of Interest cont.

- **Traffic Management Information**
  - Traffic Management Restriction
    - Data elements available for all TMIs
      - Unique ID
      - Start / End times
    - Miles in Trail
      - Spacing (NM)
      - Applicable airport / fix
    - Minutes in Trail
      - Spacing (minutes)
      - Applicable airport / fix
    - Approval Request (APREQ) List
      - Applicable airport / fix
    - Airport Departure Stop
      - Impacted NAS element
      - Reason for stop

- **Airport Information**
  - Airport Configuration
    - Arrival Runway
    - Departure Runway
  - Runway Configuration
    - Departure Rate
    - Arrival Rate
    - Runway Closure

Why NASA TTP

- **Practice**
  - NASA TTP was built against the TFDM TTP design standard
  - Using the NASA TTP provides users with a period of time to become familiar with the TTP schema and information provided
- **Integration**
  - Data generated by NASA TTP is accurate and will be similar to the data produced by TFDM
  - Users are able to begin integration of TFDM TTP data into their internal systems / operations prior to TFDM going operational
- **Feedback**
  - Using existing forums (CDM WG, SWIFT, etc.) users are able to ask questions and provide feedback to TFDM prior to its deployment

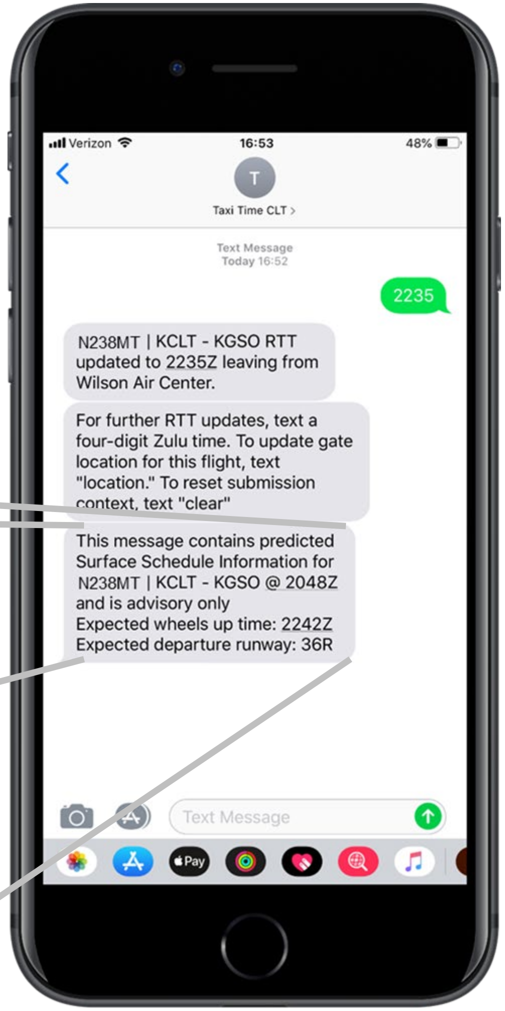
Example of TTP Utility

- **TMI Flight Lists**
  - Each TMI is published with a unique ID
    - CFR
    - Departure MIT/MINIT restrictions
    - Departure Stop
  - Flight messages published for flights impacted by a TMI(s) have the impacting TMI ID(s) included in their Flight Messages
  - Provides information needed to determine which flights are impacted by a specific TMI

Example of TTP Utility cont.

- **Airport Configuration**
  - Predicted Departure Runway
    - Flight messages published for each flight providing a predicted departure runway
    - Prediction generated by STBO model
  - Estimated Time of Departure
    - Flight messages published for each flight providing a predicted time of departure
    - Prediction generated by STBO model

MITRE Prototype using TTP data @ CLT

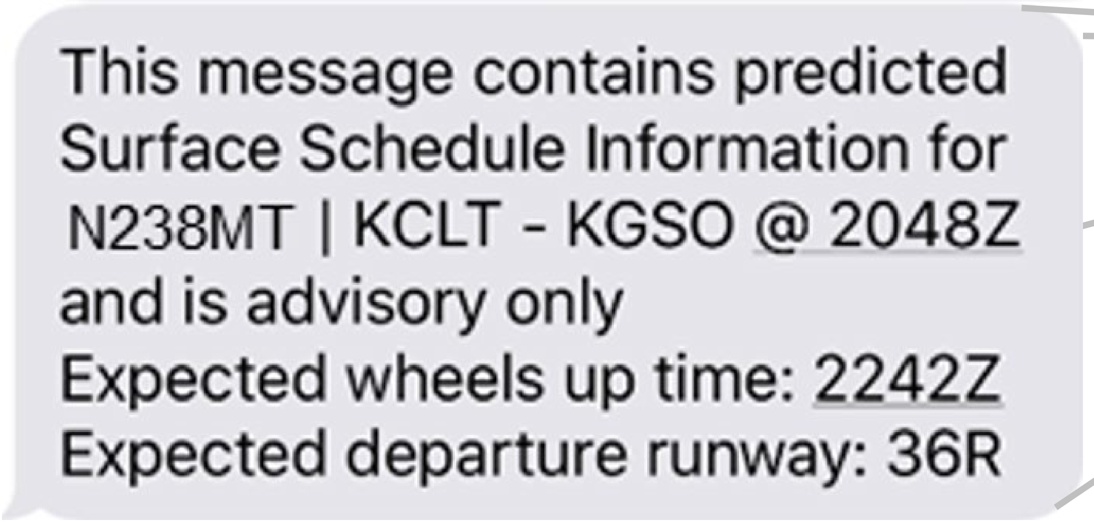


2235

N238MT | KCLT - KGSO RTT updated to 2235Z leaving from Wilson Air Center.

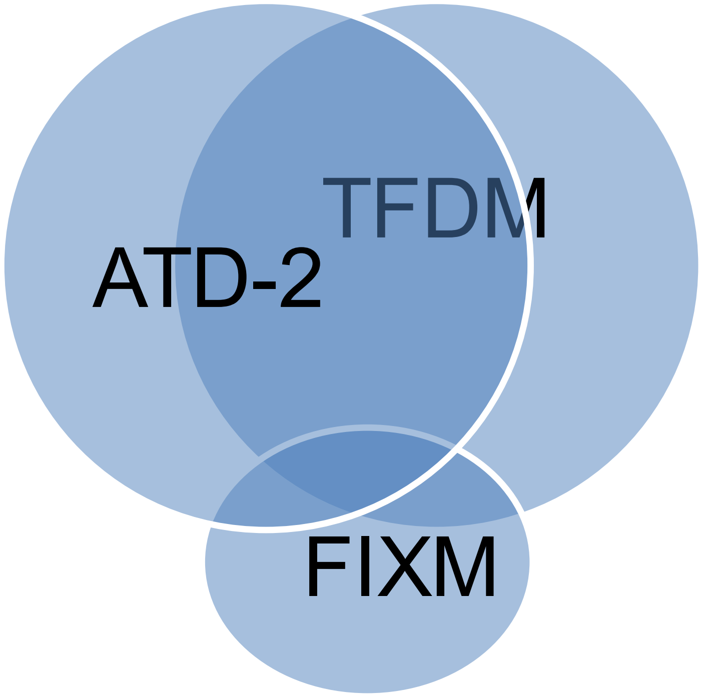
For further RTT updates, text a four-digit Zulu time. To update gate location for this flight, text "location." To reset submission context, text "clear"

This message contains predicted Surface Schedule Information for N238MT | KCLT - KGSO @ 2048Z and is advisory only
Expected wheels up time: 2242Z
Expected departure runway: 36R



This message contains predicted Surface Schedule Information for N238MT | KCLT - KGSO @ 2048Z and is advisory only
Expected wheels up time: 2242Z
Expected departure runway: 36R

Limitations



- **Program intersection limitation**
  - NASA ATD-2 has data that is not in the TFDM requirements
  - NASA ATD-2 does not have all the data to fill the TFDM requirements.
  - TFDM is expected to produce all flight data in FIXM format
  - FIXM does not currently support everything TFDM will need to publish
- **Not a one stop shop**
  - TTP generally not intended to include data that is found in other feeds

How to access ATD-2 TTP feed

- Work with SWIM to establish a connection to SWIM R&D if you don't already have a connection
  - If you already have a connection getting access to TTP will be pretty straight forward.
- Subscribe to SWIM R&D TTP feed via a new queue that will be established for each stake holder
- Work with ATD-2 team on how to utilize the information
  - See TTP Resources slide for links to documentation

TTP Resources

- **Links to FAA TFDM resources**
  - **Concept Overview:**
    - https://www.faa.gov/air_traffic/technology/tfdm/
  - **SWIM On-Ramping:**
    - https://www.faa.gov/air_traffic/technology/swim/products/get_connected/
  - **Implementation Roadmap:**
    - https://www.faa.gov/air_traffic/technology/tfdm/implementation/
- **Links to ATD-2 TFDM / NASA TTP Resources**
  - **NSRR:**
    - https://nsrr.faa.gov/services/nasa-ttp/profile
  - **NASA Website:**
    - https://www.aviationsystemsdivision.arc.nasa.gov/research/atd2/index.shtml

Questions

Where Are We?
Where Are We Going?

Where Are We Now

- Initial documentation available
  - https://aviationsystems.arc.nasa.gov/atd2-industry-days/fuser/ATD-2-Industry-Day-Documentation-Outline_81565170.html
  - Will continue to update and enhance
  - Feedback is welcome

- Collaborating
  - SWIFT: We are here to share but also to listen
  - Fuser in the Cloud
  - Help us to help you!

Fuser in the Cloud (Pilot)

- Fuser currently running in the cloud
- Cloud Benefits
  - Reduce Cost by reducing ..
    - Development time
    - Operating cost
    - Maintenance
    - Enhancements
- Delta volunteered to be our first pilot user
- Willing to engage with others that want to **partner**

5/22/2019

Where Are We Going

- Exploring various options for tech transfer
  - Knowledge transfer
  - Documentation
  - Potential software
- Exploring long term hosting options
  - Could be the answer to software challenges
- Looking for industry input to help scope the transfer
- Timeframe – 2019