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?

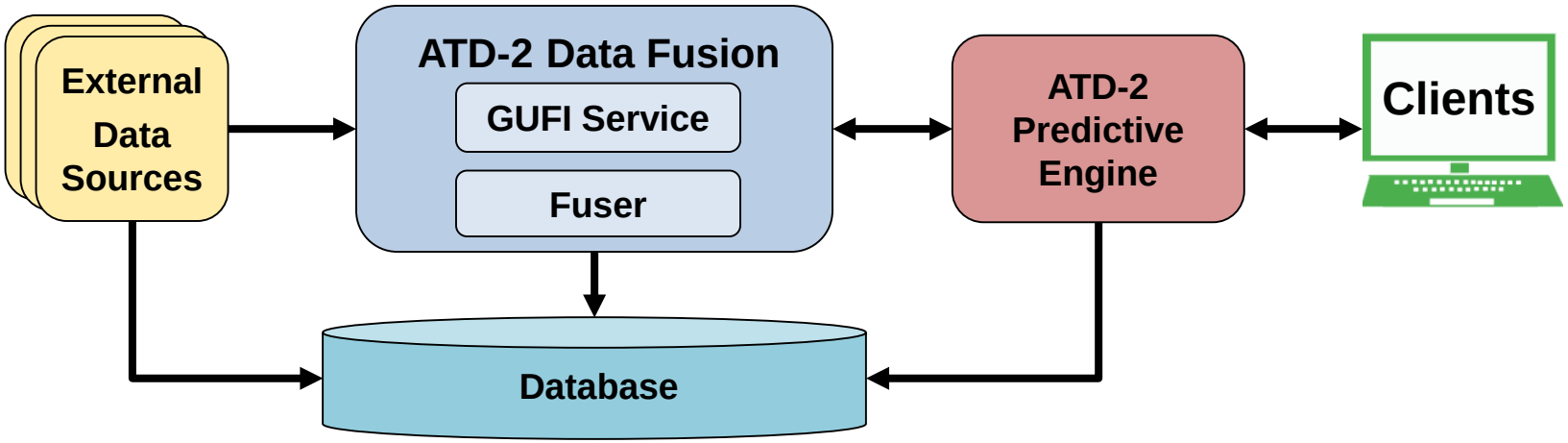
Traceability / Debugging

How did our data processing handle this case?

What triggered this update in our system?

Database Overview

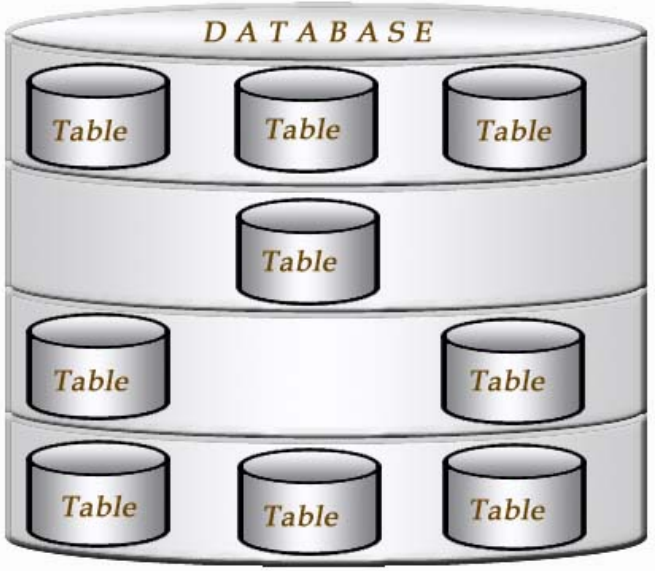
- Each ATD-2 system has a dedicated system that captures the data necessary to meet objectives
- Each database contains tables with
  - Flattened messages from external sources
  - Flight matching data
  - Fuser mediated data
  - Data from ATD-2 internal components
- Database structure prefers wide tables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

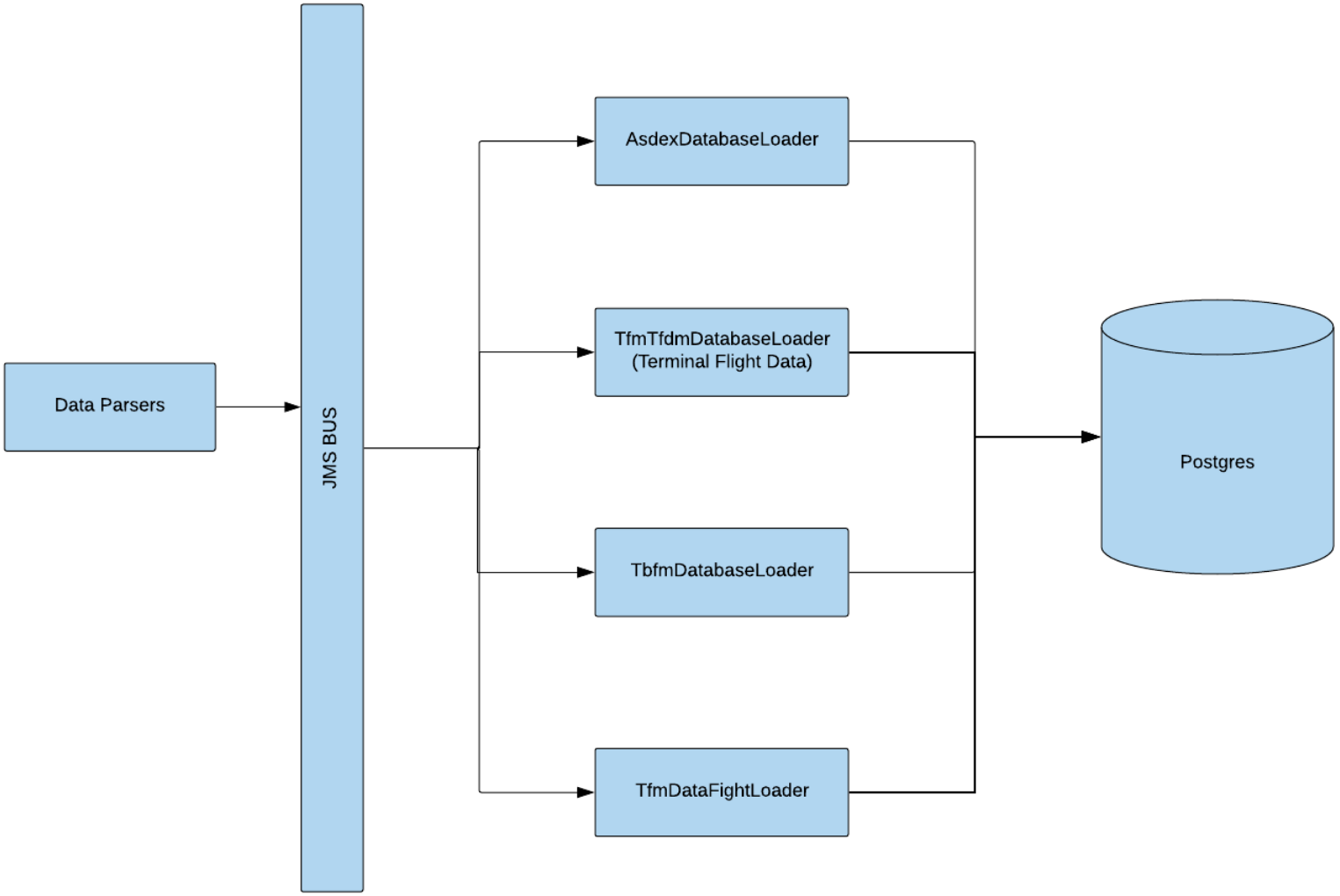
tma_message	NASA research TBFM and SWIM TBFM data
	• See the source column. Contains either
	  • SWIM
	  • NASA-ZTL
	  • NASA-ZDC

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

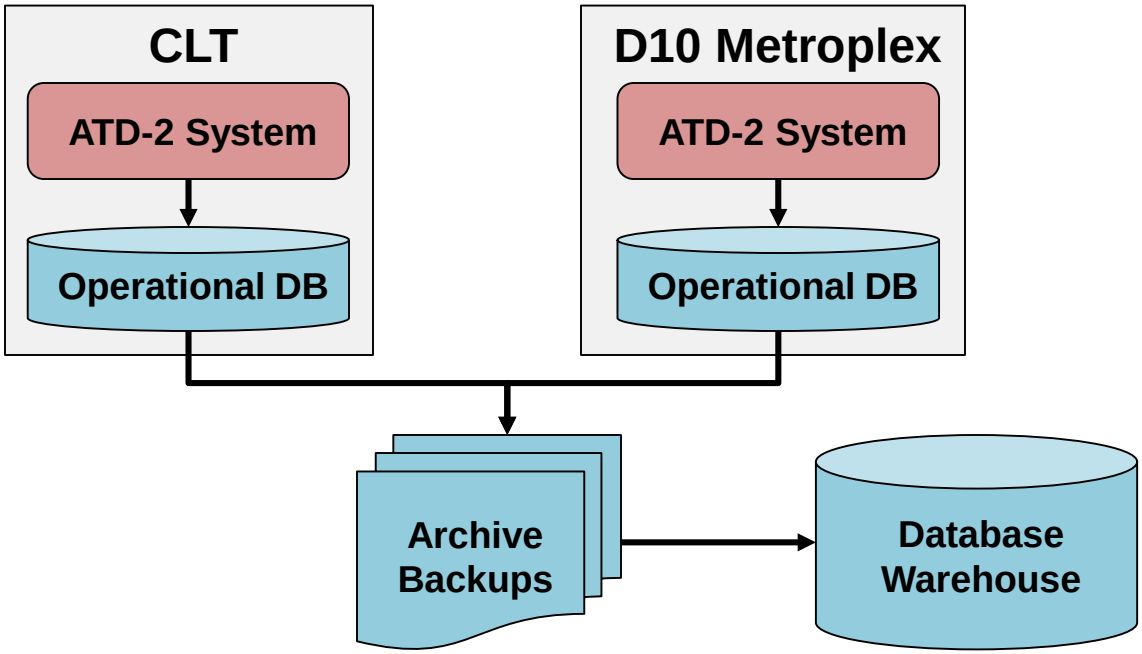
- All tables are partitioned based on timestamp
  - Improves query performance
  - Allows old data to be easily rolled off
  - Data is automatically rolled off daily
  - Data is stored long term in a warehouse

- 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

```
<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

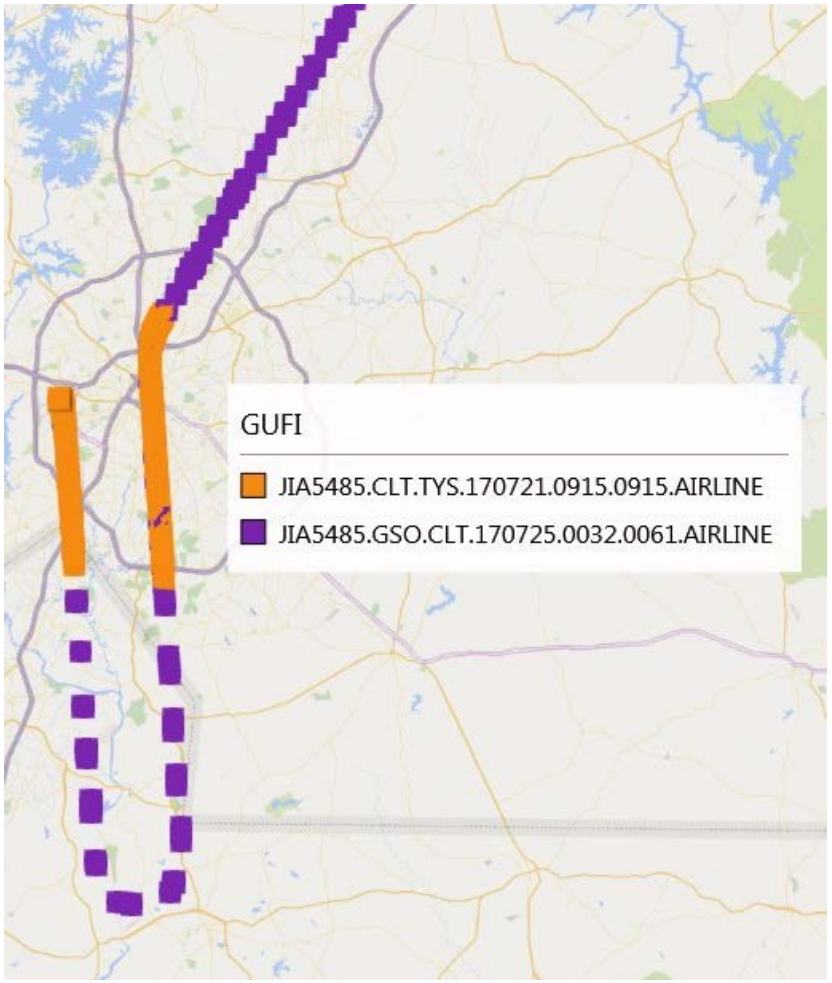
Use Case: Traceability / Debugging
Flight Matching Issues

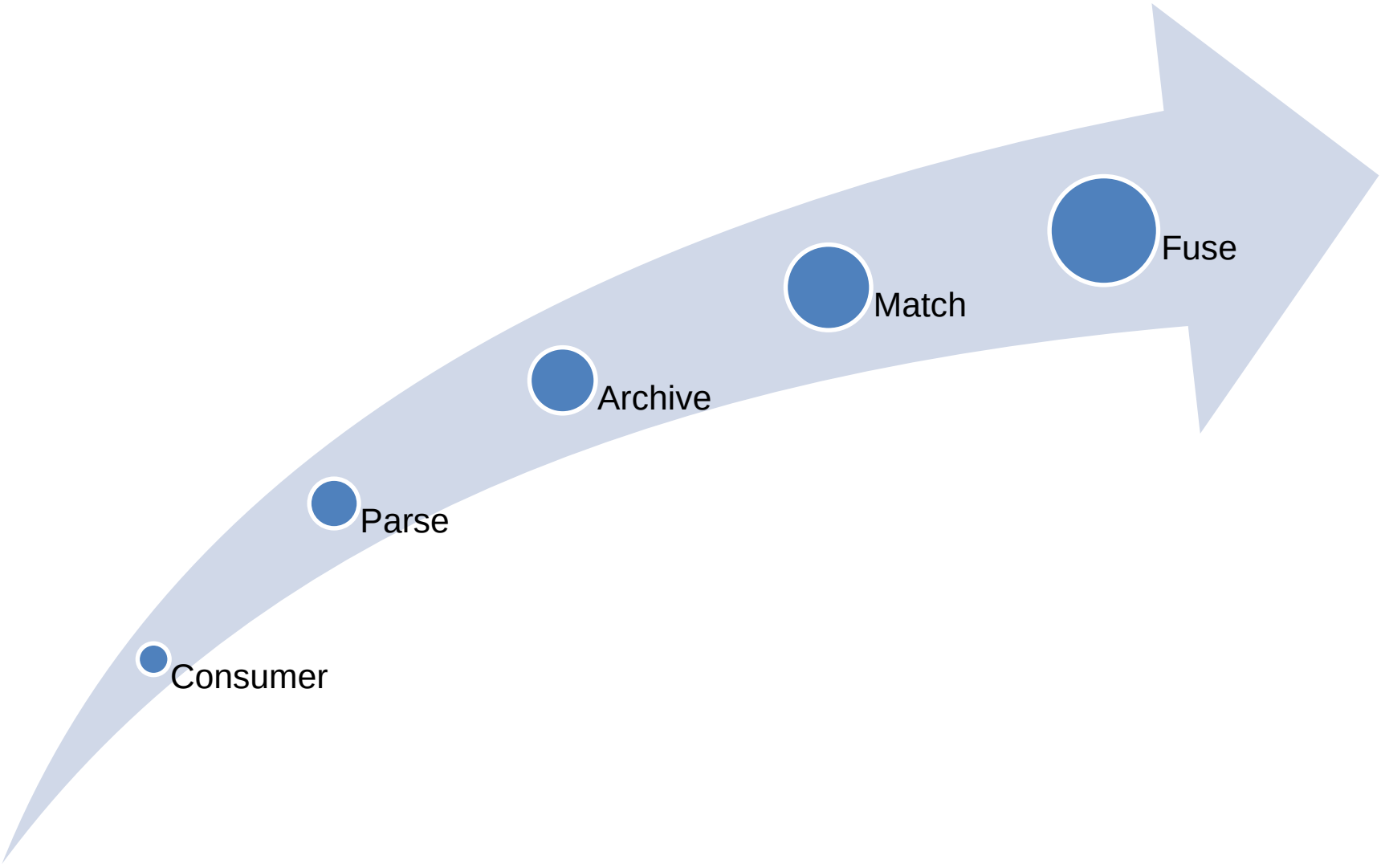
- Question:
  - Prior to going live at CLT, there was a large push to resolve flight matching issues
  - Testers reported an issue where two aircraft icons were observed for a single arrival
  - What happened?

- 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

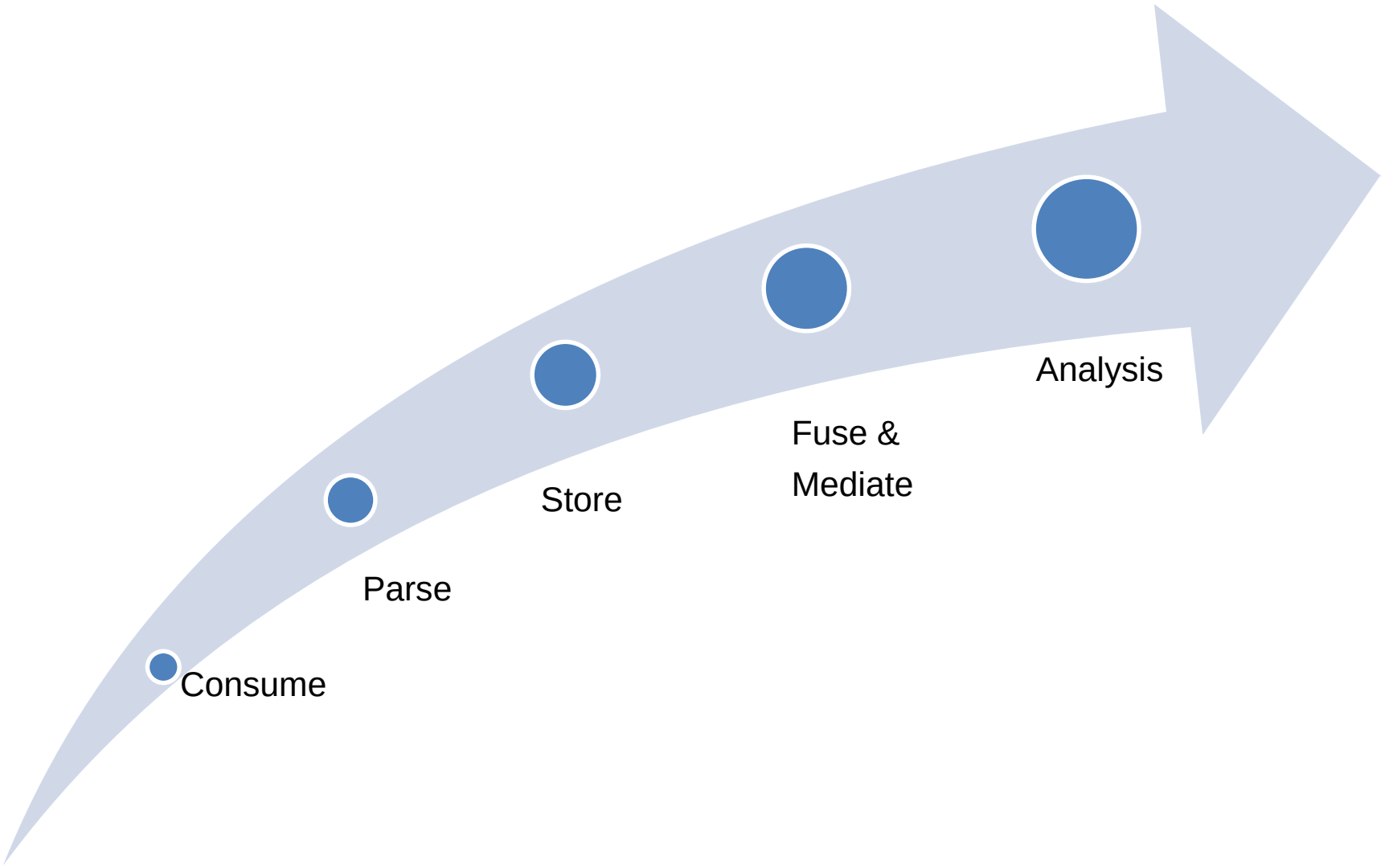
Use Case: Traceability / Debugging
Flight Matching Issues

- 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

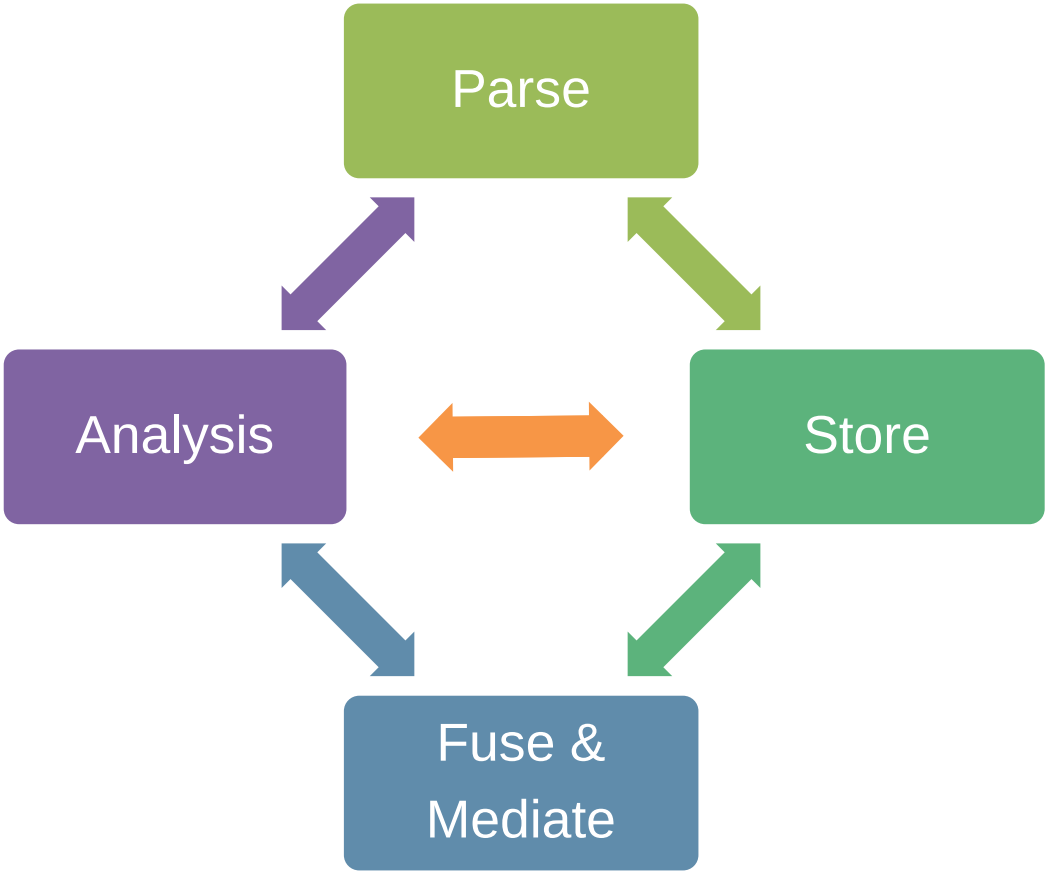



Fuse
Match
Archive
Parse
Consumer

Planned Roadmap



Actual Roadmap



Questions