SWIM Interactive Developer Workshop 2016

Introduction to SWIM Flight Data Publication Service (SFDPS)

By: George Curley
Volpe Center, SFDPS Development

Date: June 22, 2016

Federal Aviation Administration

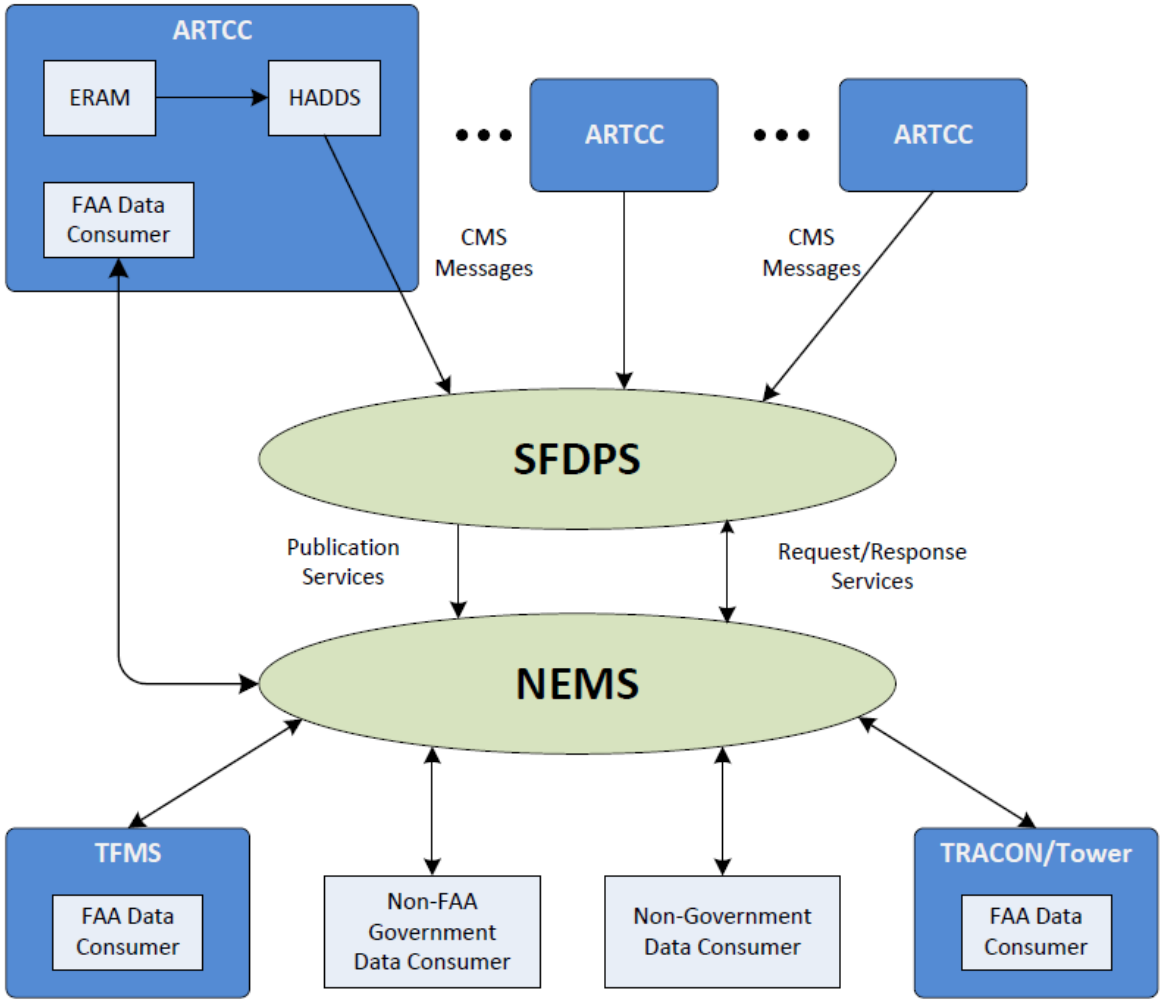
SWIM
Interactive Developer
Workshop 2016

What is SFDPS?

- SFDPS stands for SWIM Flight Data Publication Service
- SFDPS is the SWIM interface for ERAM

- SFDPS producer services include:
  - Publish/Subscribe services
  - Request/Response services
- SFDPS publishes four types of messages
  - Flight Messages in Simple XML or FIXM XML format
    - All flight messages correlated through an SFDPS flight identifier
    - Flight track update messages:
      - Currently restricted to one-minute frequency
  - Air Space messages in Simple XML or AIXM XML format
  - Operational messages
  - General Information messages

SFDPS – Operational View



SFDPS Highlights

- SFDPS is an early adaptor of both FIXM and AIXM standards
- Data available through both Publish/Subscribe and Request/Response
- Current and historical databases
  - Current database represents active and proposed flights
  - Historical is a rolling15 days
    - Retrieve messages from past 15 days.
- Flight matching through a unique flight identifier
- Data Reconstitution for:
  - Flights (active and proposed)
  - Sector (current sectorization for centers)
  - Routes (current route configurations)

SFDPS Consumers

- SFDPS supports many types of consumers:
  - FAA internal, operational consumers; for example:
    - TFMS or TBFM
  - FAA internal, non-operational consumers; for example:
    - Data Archival or Quality Assurance functions
  - External, operational consumers; for example:
    - Airline system operation centers
    - Military and other non-FAA government organizations.
    - International air traffic management systems
  - External, non-operational consumers; for example:
    - Research organizations

SFDPS value proposition for Client Developers

SFDPS provides data in two different formats:

Simple XML

Canonical , simple , one-to-one mapping to ERAM data. Binary Messages from HADDS are transformed to XML format, and additional useful fields are added with the SFDPS Flight matching capability.

FIXM XML

SFDPS also produces data in FIXM format, helping users to conform to the leading international standard for Flight data. The Flight Information Exchange Model (FIXM) is a data interchange format for sharing information about flights throughout their lifecycle. FIXM increases interoperability among all air traffic stakeholders.

SIMPLE XML Schema

Snippet of Simple XML schema shown here:

Meta-data

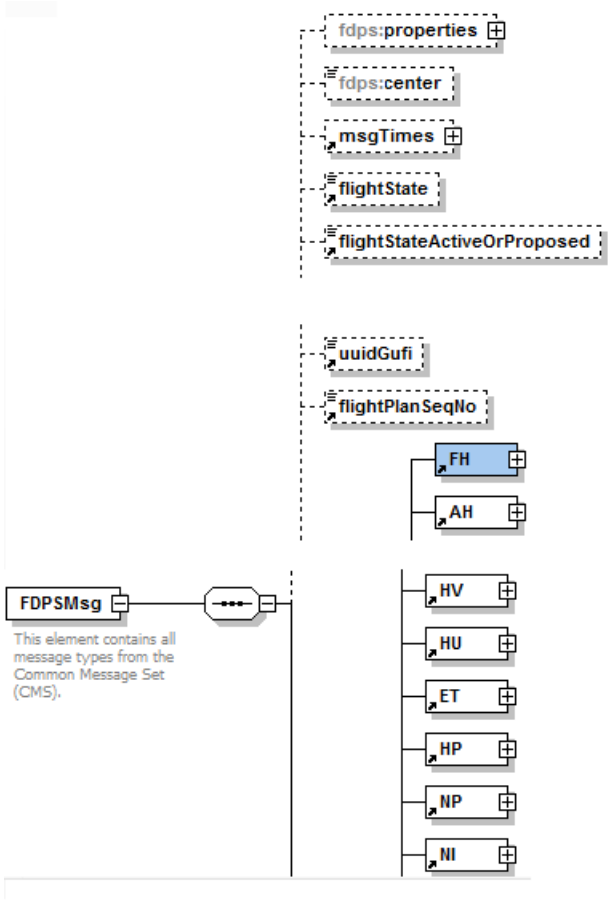
This is value-added data filled in by SFDPS, after flight matching, and includes several fields that would be useful to users.

Examples. OriginAirport, DestAirport, UUID GUFI

CMS Messages

All 40 ERAM/HADDS messages are included In the SFDPS Schema.

Examples: FH, AH, TH, DH.



FIXM XML Schema

Snippet of FIXM XML schema shown here:

Flight Identification

Identifying information for a flight

Flight Status

A set of status indicators for the current flight

GUFI

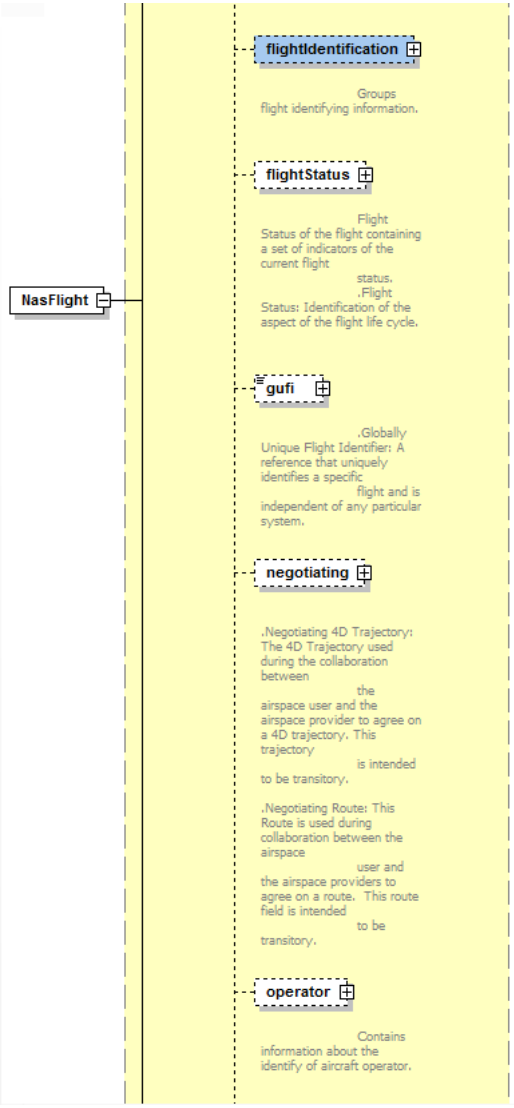
Globally Unique Flight Identifier

Negotiating

Negotiating 4D trajectory between Airspace User an

Operator

Aircraft Operator Identifying Information



SFDPS value proposition for Client Developers

SFDPS provides an “enriched” data set due to its sophisticated flight matching capabilities.

What is Flight Matching?

Every message flowing through SFDPS is matched in the SFDPS database, to compare it to existing NAS flight information, and to determine key identification information for the flight . This helps SFDPS to uniquely identify, track and enrich incoming ERAM messages, with fields that will be valuable to users.

As an example: let’s consider the “life” of a Position (TH) message as it flows through SFDPS to see how it is enriched.

SFDPS value proposition for Client Developers

A moment in the life of a TH message

A "story" illustrating how flight matching enhances data fields inside a Position (TH) message, as it flows through SFDPS

SFDPS value proposition for Client Developers

What happens to a TH message as it flows through SFDPS?

Step 1: Convert the TH coming in from ERAM/HADDS from binary to ASCII format. This is a one-to-one mapping of fields.

Step 2: Convert ASCII message to Simple XML format.

Step 3: Match the message to corresponding flight records in the database and fill in the metadata and useful identifying information including the UUID GUFI, ETA, ETD, Origin airport, Destination airport, etc.

Step 4: Convert message to FIXM format.

Step 5: Fill in “routable” JMS properties that NEMS can use to filter and route messages with, and publish both SIMPLEXML and FIXM format of TH message to users via NEMS. For example, a user interested in getting TH messages from KBOS can configure NEMS to receive only these messages. This would not be possible without flight matching.

SFDPS value proposition for Client Developers

What happens to a TH message as it flows through SFDPS?

TH Message coming in to SFDPS from HADDS is in Binary format

SFDPS value proposition for Client Developers

```
2A0     DAL1230         2D0     840
        167A0   429     5B0     479
8A0     320     54A0    314     54B0
        T       138A0   ZMP     138B0
17      23D0    425226N/0924128W
        23E0    -271/+396       170A0
2013-07-14T16:28:09Z  171A0
        425215N/0924133W        172A0
314 173A0 2013-07-14T16:28:09Z
```

Binary format message is first converted to ASCII format (one to one field mapping)

SFDPS value proposition for Client Developers


```
<singleTH>
  <THmetadata>
    <propSourceSystem>DevCloud</propSourceSystem>
    <propFlightId>DAL1230</propFlightId>
    <propFlightOperator>DAL</propFlightOperator>
    <propOrigin>KATL</propOrigin>
    <propDestination>KMSP</propDestination>
    <propRcvdTime>2016-06-16T16:28:15.118Z</propRcvdTime>
    <propRcvdTimeEpoch>1466094495.118474</propRcvdTimeEpoch>
    <propSentTime xsi:nil='true'></propSentTime>
    <propSentTimeEpoch xsi:nil='true'></propSentTimeEpoch>
    <propSeqNo>0000006066</propSeqNo>
    <propTestMsg xsi:nil='true'></propTestMsg>
    <propOneMinFreq>true</propOneMinFreq>
    <msgTimes>
      <arrivalTime>2013-07-14T16:52:00Z</arrivalTime>
      <arrivalTimeEpoch>1373820720000</arrivalTimeEpoch>
      <departureTime>2013-07-14T16:06:00Z</departureTime>
      <departureTimeEpoch>1373817960000</departureTimeEpoch>
    </msgTimes>
    <flightState>Active</flightState>
    <flightStateActiveOrProposed>true</flightStateActiveOrProposed>
    <fdpsGufi>us.fdps.2016-06-16T15:13:25Z.000/10/100</fdpsGufi>
    <eramGufi_316aFPId>KK54805100</eramGufi_316aFPId>
    <eramGufi_316aDT>2016-06-16T15:13:25Z/000/10/100</eramGufi_316aDT>
    <uuidGufi>c0b28bab-ac7b-41b8-b15c-13b19cc41e2d</uuidGufi>
    <flightPlanSeqNo>6</flightPlanSeqNo>
  </THmetadata>
  <TH>
    <sourceId_00e>1628138965</sourceId_00e>
    <sourceTime_00e1>16_28_13</sourceTime_00e1>
    <sourceSeqNo_00e2>8965</sourceSeqNo_00e2>
    <flightId_02a>DAL1230</flightId_02a>
    <computerId_02d>840</computerId_02d>
    <sspId_167a>429</sspId_167a>
    <groundSpeed_05b>479</groundSpeed_05b>
    <assignedAlt_08a>320</assignedAlt_08a>
    <reportedAlt_54a>314</reportedAlt_54a>
    <reportedAlt_54b>T</reportedAlt_54b>
    <controllingFacility_138a>ZMP</controllingFacility_138a>
    <controllingSector_138b>17</controllingSector_138b>
    <trackPosition_23d>425226N/0924128W</trackPosition_23d>
    <trackVelocity_23e>-271/+396</trackVelocity_23e>
    <timeOfTrackData_170a>2013-07-14T16:28:09Z</timeOfTrackData_170a>
    <targetPosition_171a>425215N/0924133W</targetPosition_171a>
    <targetAlt_172a>314</targetAlt_172a>
    <timeOfTargetData_173a>2013-07-14T16:28:09Z</timeOfTargetData_173a>
  </TH>
</singleTH>
```


SFDPS generates SIMPLE XML format TH message, after flight matching. This results in several new and useful fields being added to the original TH message including: Origin Airport, Destination Airport, UUID GUFI

SFDPS value proposition for Client Developers


```
<message xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ns5:FlightMessageType">
  <flight xsi:type="ns5:NasFlightType" centre="ZAU" source="TH" system="DevCloud" timestamp="2016-06-16T16:28:15.118Z">
    <arrival xsi:type="ns5:NasArrivalType" arrivalPoint="KMSP">
      <runwayPositionAndTime>
        <runwayTime>
          <estimated time="2013-07-14T16:52:00Z"/>
        </runwayTime>
      </runwayPositionAndTime>
    </arrival>
    <controllingUnit xsi:type="ns4:IdentifiedUnitReferenceType" unitIdentifier="ZMP" sectorIdentifier="17"/>
    <departure xsi:type="ns5:NasDepartureType" departurePoint="KATL">
      <runwayPositionAndTime>
        <runwayTime>
          <actual time="2013-07-14T16:06:00Z"/>
        </runwayTime>
      </runwayPositionAndTime>
    </departure>
    <enRoute xsi:type="ns5:NasEnRouteType">
      <position xsi:type="ns5:NasAircraftPositionType" targetPositionTime="2013-07-14T16:28:09Z" positionTime="2013-07-14T16:28:09Z" reportSource="SURVEILLANCE
        <actualSpeed>
          <surveillance uom="KNOTS">479.0</surveillance>
        </actualSpeed>
        <altitude uom="FEET">31400.0</altitude>
        <position xsi:type="ns4:LocationPointType">
          <location srsName="urn:ogc:def:crs:EPSG::4326">
            <pos>42.873889 -92.691111</pos>
          </location>
        </position>
        <targetAltitude uom="FEET">31400.0</targetAltitude>
        <targetPosition srsName="urn:ogc:def:crs:EPSG::4326">
          <pos>42.870833 -92.6925</pos>
        </targetPosition>
        <trackVelocity>
          <x uom="KNOTS">-271.0</x>
          <y uom="KNOTS">396.0</y>
        </trackVelocity>
      </position>
    </enRoute>
    <flightIdentification xsi:type="ns5:NasFlightIdentificationType" computerId="840" siteSpecificPlanId="429" aircraftIdentification="DAL1230"/>
    <flightStatus xsi:type="ns5:NasFlightStatusType" fdpsFlightStatus="ACTIVE"/>
    <gufi codeSpace="urn:uuid">c0b28bab-ac7b-41b8-b15c-13b19cc41e2d</gufi>
    <operator>
      <operatingOrganization>
        <organization name="DAL"/>
      </operatingOrganization>
    </operator>
    <supplementalData xsi:type="ns5:NasSupplementalDataType">
      <additionalFlightInformation>
        <nameValue name="MSG_SEQ_NO" value="6066"/>
        <nameValue name="FDPS_GUFI" value="us.fdps.2016-06-16T15:13:25Z.000/10/100"/>
        <nameValue name="FLIGHT_PLAN_SEQ_NO" value="6"/>
      </additionalFlightInformation>
    </supplementalData>
    <assignedAltitude>
      <simple uom="FEET">32000.0</simple>
    </assignedAltitude>
    <flightPlan identifier="KK54805100"/>
  </flight>
</message>
```


SFDPS also generates FIXM XML format TH message which conform to the International FIXM standard, and also includes the additional useful fields such as : Origin Airport, Destination Airport, UUID GUFI, etc

SFDPS value proposition for Client Developers

SFDPS also provides data via Web Services in Request/Response mode:

Current Data Web Service Requests

Using standard SOAP services users can request the "current" state of a single flight or the entire NAS:

Examples:

- *Give me the current flight plan and position update for Flight XXX*
- *Give me detailed information for ALL active and proposed flights*
- *Give me the current sector configurations*

Historical Data Web Service Requests

Users can request historical data for up to 15 days using standard SOAP web services.

Examples:

- *Give me all flight plans filed from KBOS to KLAX for the last 7 days*
- *Give me all flights originating from KLAX for the last 10 days*

What's Coming Next in SFDPS?

12-second position updates!

- SFDPS will make 12-second postion updates available. Position messages (TH) will be batched
- This will require users to update their clients
  - SFDPS user forums are being provided to facilitate the transition
  - SFDPS has provided through the NSRR:
    - ❖ SFDPS Connect client source code
    - ❖ Video training tutorials on the SFDPS Connect client

Questions/Comments?

Backup Slides

FAQ

What advantages does SFDPS offer users:

- FIXM and AIXM formats:
  - All FAA and worldwide systems planned to eventually migrate to these standards
- An enhanced “cleaned up” data feed:
  - Eliminates the meaningless messages, such as cancellation messages for airborne flights
  - Eliminates redundant or conflicting messages, such as flight plans and track messages from non-controlling centers
- Value added data fields:
  - A unique flight identifier that can be used to match messages regardless of what center they come from
  - Flight status (for example, planned, active, cancelled, completed)
  - Sequence numbers, so that you reorder data (if needed) and identify gaps

FAQ

- What will SFDPS include that TFM_Data doesn't?
  - SFDPS has additional route fields (filed/requested and ATC intended) in addition to what route is cleared
  - Flight rules and equipment codes extracted from the ICAO flight plan
  - More frequent track updates (12-second)
  - Hold information

FAQ

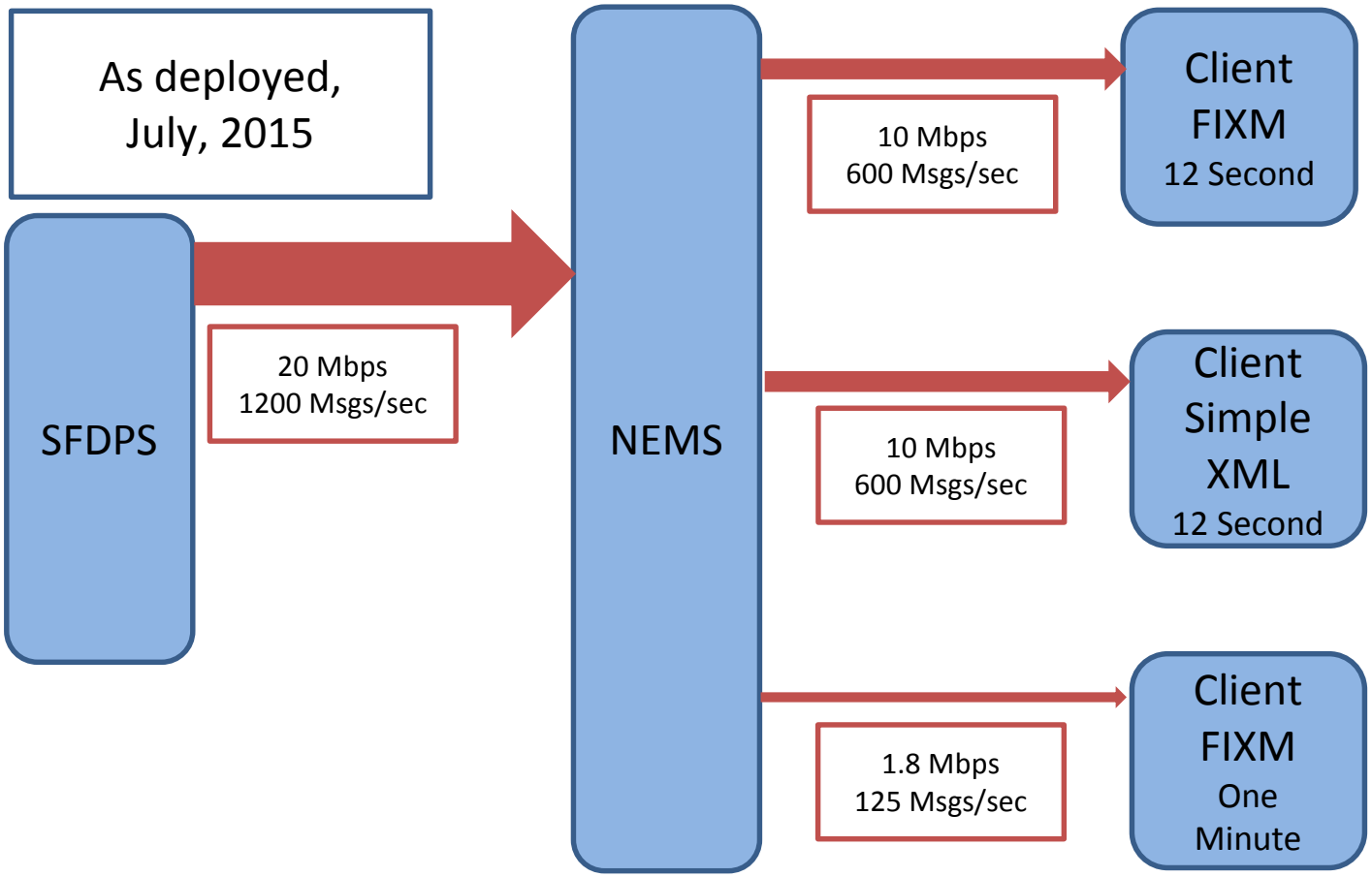
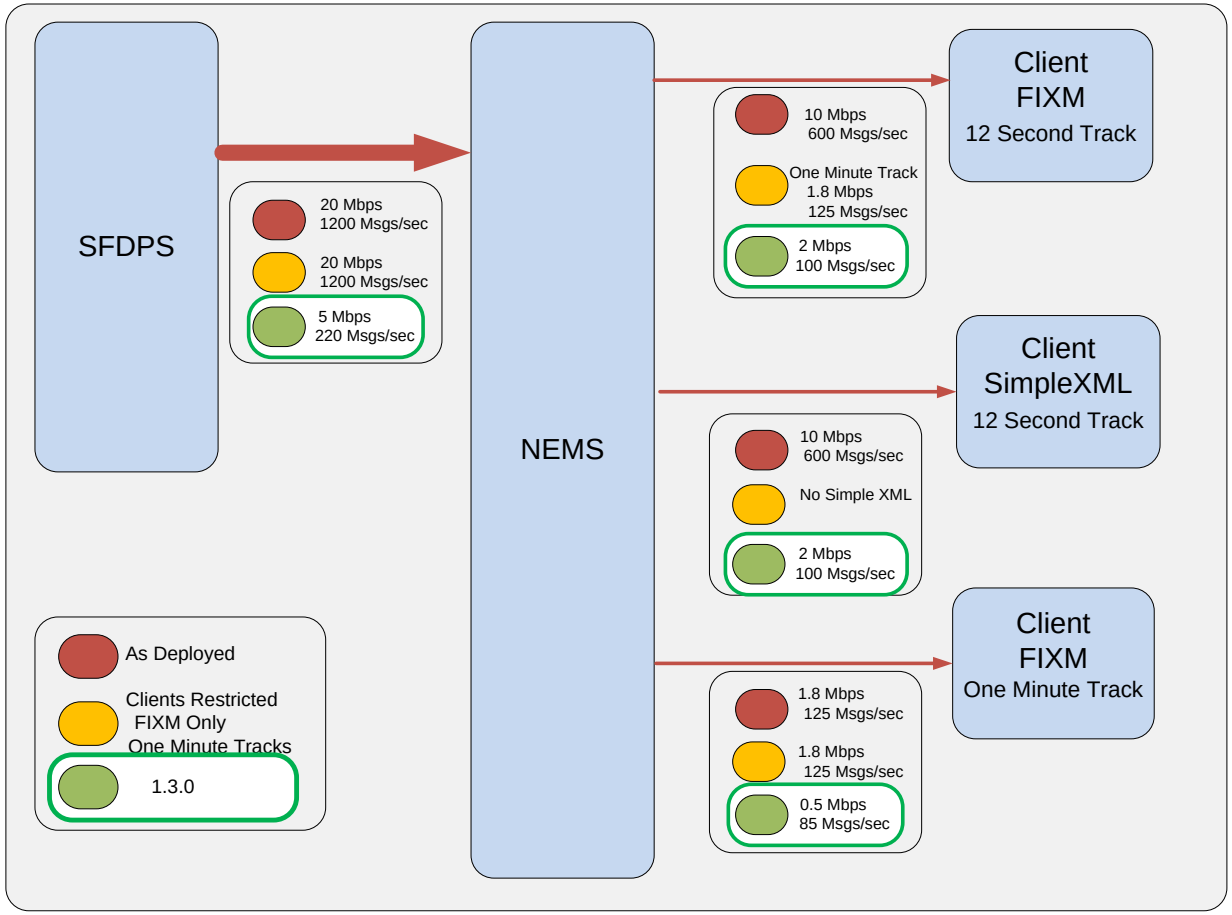
What advantages does SFDPS offer? (continued)

- SFDPS provides flexible data filtering; that is, you can get only data that meets selected criteria
  - Message type
  - ARTCC
- You can make data requests:
  - If your system starts up new or recovers from a failure, it can request data to immediately repopulate its database
  - If you are not sure you have the latest data about a flight or a route status, for example, you can request just that data

SFDPS version 1.3.0

- SFDPS 1.3.0 solves the problem of excessive message rate and bandwidth usage by batching track messages (TH)
- SFDPS 1.3.0 with batching enabled will:
  - Substantially reduce message rates and bandwidth usage
    - Achieve levels comparable to the current, restricted flow
  - Remove the need for a restricted data feed
    - Restore the 12-second track updates

Peak Bandwidth and Message Rates



Peak Bandwidth and Message Rates Summary