



Use of AMQP in the SESAR MET-GATE project

Yann Génin (DSI/DEV)
05/03/2020

Agenda

- The MET-GATE project
- First step : introduction of AMQP in the MET-GATE project
 - Why AMQP ?
 - How it was used
 - The SWIM Global Demonstration : lessons learned
- Second step : toward an operational system
 - New context : the MF MET-GATE project
 - AMQP : how it will now be used
 - Future work / opportunities



- Project launched in 2012 within SESAR program

- First step : to build the MET-GATE concept
 - To define new services delivering MET information to Aviation users
 - To demonstrate these new services in support to validation exercises

- Second step (2016 - 2020): to build a pre-operational service based on previously developed concept
 - To consolidate the concept
 - To implement a resilient system

First step : introduction of AMQP in the MET-GATE project

- The need : to push MET information to users as soon as it is available
 - Users subscribe to MET products via a Webservice
 - Users get notifications via an AMQP queue
 - Different size of data, between 10 B and 10 GB
 - Small messages : METAR, TAF, SIGMET...
 - Large model output grids

Why use AMQP ?

- Requirement : the MET-GATE must comply with the SWIM Yellow Profile
- 2 options for user notifications :
 - WS-Notification (SOAP)
 - AMQP

Why use AMQP ?

- WS-Notification
 - SOAP defines subscription interfaces
 - Homogeneous stack of technologies
 - More than 100 different requirements
 - Which ones will apply ??
 - SOAP isn't lightweight
 - Xml verbosity
 - Interoperability depends on implementation
- AMQP
 - Alternative to SOAP
 - Newer than SOAP
 - Wire-level protocol :
 - messages can use any format
 - XML was chosen
 - A single requirement :
 - Use AMQP 1.0 .
 - The dev team had experience with JMS
 - Concepts of queues, topics, ...
- Project choice : AMQP 1.0

AMQP : How it was used (first step)

- AMQP implementation : ActiveMQ
- Notifications :
 - XML messages containing a URL to data published on a queue per user
 - User has to retrieve the data via http
 - Good for large piece of data (subset of a weather prediction model grid, set of objects intersecting the user's region of interest...)
 - Not very efficient for small data (OPMET eg. METAR message...) :
 - The URL may be longer than the message itself



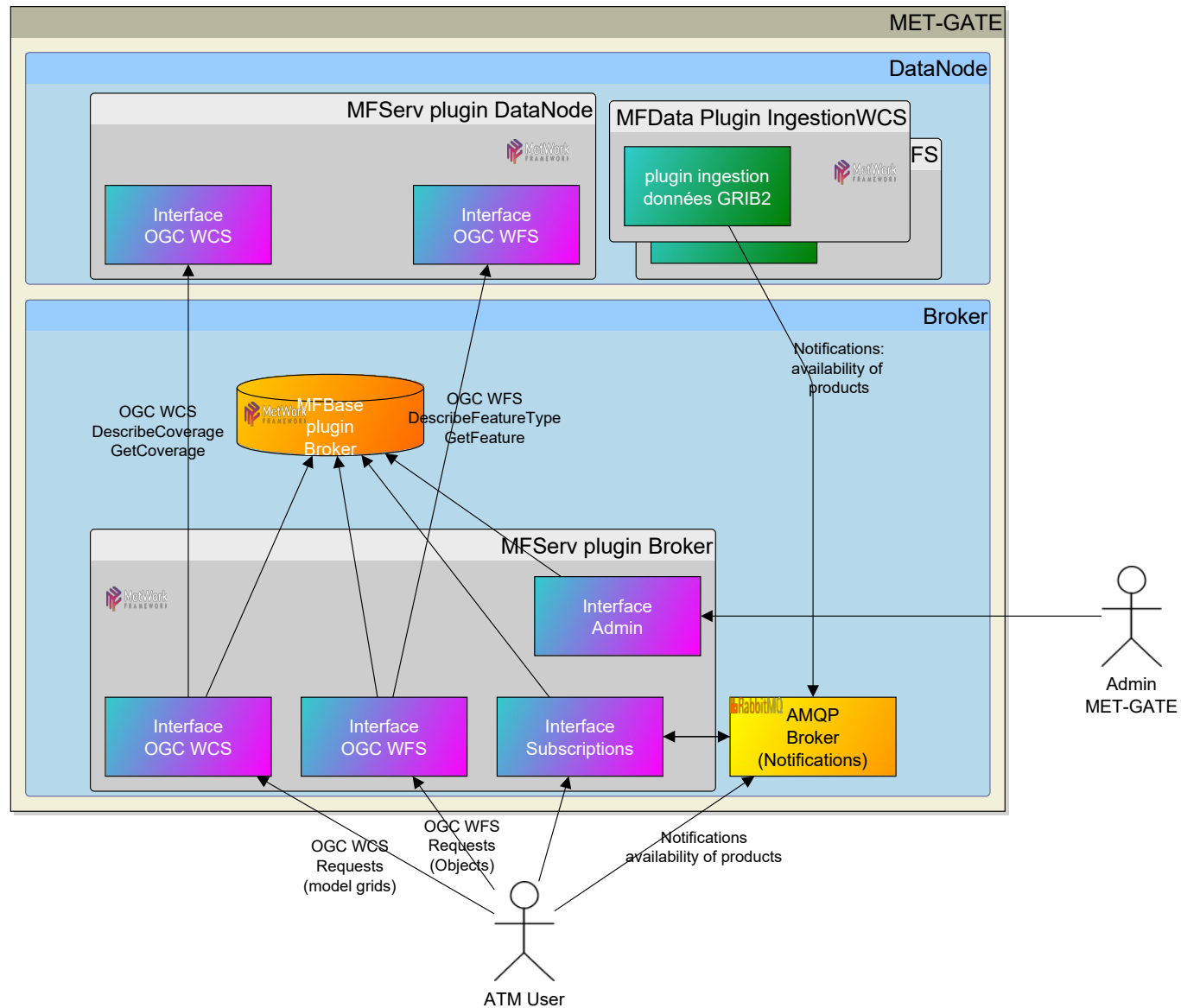
SWIM Global Demonstration : lessons learned (Rome, 2016)

- First « real life » test for the MET-GATE AMQP interface
- It worked !
 - But it wasn't easy to debug :
 - The protocol was not well supported by IT Security team
- AMQP is popular and quite well supported by developers
 - Most of the services provided AMQP interfaces
- Lack of harmonization
 - Notifications were implemented in different ways
 - Most services provided the data in the notification message (only small piece of data were exchanged)
 - No common format for notification
 - Clients had to develop specific code for each service


MET-GATE : the second step

- Goal : to setup a pre-operational system
 - Context has changed : MET-GATE is now a MF project
 - Harmonization effort to be kept with other European MET services
 - More focused on our users' needs
- New development based on METWORK Framework
(<https://network-framework.org/>)
 - Programming language : python

The MET-GATE architecture



AMQP implementation in MET-GATE prototype

- New choice : RabbitMQ  RabbitMQ by Pivotal.
 - New development team, new preferences
 - AMQP 0.9.1 used for internal interfaces
 - Python library: pika
 - Provides AMQP 1.0 interface via a plug-in : SWIM requirement is satisfied.
- Implementing guidelines were developed for the SESAR SWIM TI Yellow Profile
 - Guidance for Pub-Sub push implementation
 - Product configuration examples (including RabbitMQ)

AMQP in MET-GATE

- Notification format :
 - Short XML document
 - Notification of availability of a new dataset
 - Will include data if size is « small » enough
 - Definition of « small » to be refined :
 - Less than 100 kB ? Less than 10 kB ? 1 kB ?
 - eg. METAR , TAF ... messages in text format
 - METAR, TAF, messages in iWXXM format

- 1 queue per user
 - Messages are duplicated for each user who subscribed for a given notification type

AMQP in MET-GATE : future work, opportunities

- Rework on message format
 - Should we keep XML ?
 - Opportunity to align with work done on WIS2 ?
- Possibility to provide notifications with criteria on the data values
 - New work on subscription interfaces

AMQP in MET-GATE : future work, opportunities

- Coming challenges :
 - To set up a resilient AMQP interface (High availability)
 - To ensure that a message isn't sent twice to the same user
 - To ensure that messages are effectively delivered to users
 - Performance
 - How many concurrent users can be served simultaneously ?
 - How to scale to support more users ?
 - Interfaces :
 - Future version of the MET-GATE will be hosted behind an API manager (WSO2 APIM)
 - AMQP service should use APIM user authentication

Questions ?