

Air France-KLM Development Performance Soars with API Management Innovation



The Air France-KLM airline group's main businesses are aeronautical maintenance and air transport of passengers and freight. It is first in intercontinental traffic from Europe and a major provider of global air transport. Air France-KLM operations involve a lot more than managing flight logistics. Behind the airport and flight operations that passengers see is a huge digital API-driven business that makes daily operations possible—from customer ticket purchases to flight maintenance.

Before the COVID pandemic, the Air France-KLM Group's goal was to become the world's most customer-centric airline; it was investing heavily in passenger service innovation. But after the onset of the pandemic, the company's priority shifted to digitizing commercial cargo and freight operations. It also needed to increase business agility to adapt to rapidly and continuously changing travel restrictions.

To accomplish these goals, the company worked with TIBCO to create an innovative solution to speed API delivery using TIBCO Cloud API Management software.

Common Pain Points

In the fast-paced transportation business, it is critical to correctly deploy to each production endpoint. Air France-KLM uses 250-plus TIBCO Cloud API Management production endpoints that are consumed by more than 300 applications developed internally and by business partners. Failure of these applications can wreak havoc in daily operations:

- Aircraft push-back from the gate is delayed if the flight crew doesn't receive the passenger list and airway bill for cargo in the hold.
- If a maintenance engineer is not properly notified of service requests, an aircraft may not be returned to service on time, leading to flight delays.
- If a logistics partner is not able to book air freight automatically, the shipments may be delayed.

The centralized API management team is responsible for ensuring that the correct security policies and governance are enforced on every API endpoint created by more than 80 product teams. When a product team requests to publish a new API, multiple processes are set in motion to ensure compliance with company policies. Additionally, Air France-KLM requires specific security architectures and protocols to ensure passenger data is safe and in compliance with the General Data Protection Regulation (GDPR). Which security measures are required can vary based on the type of data being handled. It is paramount that the API management team verifies correct implementation of security guidelines to ensure that:

- All applicable security requirements have been applied.
- The solution is running on an adequately secure infrastructure.
- The TIBCO Cloud API Management configuration is in line with security guidelines.

Pandemic-related Struggles

Air France-KLM has a vast and complex IT infrastructure, including multiple on-premises and cloud data centers. To avoid downtimes and operations disruptions, the company requires that its 80+ product teams adhere to the CI/CD lifecycle. Planning and executing a successful API rollout requires deep knowledge of all eight environments available for continuous integration and test.

Based on experience, the API management team found that most product teams requested API deployments with higher security requirements than were necessary. This overcomplicated the API deployment process, leading to greatly increased workloads. Before COVID, it could take the API management team a full calendar week to complete the necessary deployment setup. This process could have been shortened to several hours if a simpler alternative was chosen.

Additionally, a product team could lose track of what the security measures accomplish, leading to outages regardless of the testing throughout the development lifecycle. At Air France-KLM, several acute traffic interruptions occurred due to changes made to the security constructs. The worst event deleted security credentials for more than 30 applications.

These credentials could not be restored and had to be reissued, resulting in an application downtime of approximately three business days.

To minimize the chance for disruptions, the API management team implemented a specialist peer review process at multiple points in the API development journey. Prior to the pandemic, the API management team had 12 specialists covering technical review including change management, API management administration, advanced networking, and programming. After pandemic downsizing, the smaller API management team of three specialists struggled to execute deployments correctly, creating a greater risk of outages that could result in lost sales and delays in airline operations.

As economies reopened and travel resumed, the team was under pressure to shorten lead times for API publication to meet project timelines. The circumstances demanded that the remaining team quadruple efficiency to keep up with demand.

Working Smarter, Not Harder

To address its challenges, Air France-KLM took a critical look at processes and tooling to reduce the processing time for creating or modifying automated deployments. The company identified two prime areas for improvement:

- Creating a feedback mechanism to identify incorrect interpretations of the Air France-KLM security guidelines was needed. In the previous process, mistakes were identified during user acceptance testing after several environments had already been configured.
- Training new employees in deployment processes had a steep learning curve:
 - The XML deploy-based descriptors used in the API endpoint configuration (now known as Digital.ai) were specific to Air France-KLM, meaning new employees had to start from scratch even if they had prior devOps experience.
 - After new employees learned the descriptor format, it then took considerable time to understand how the API landscape translated into values entered in the deployment descriptor. This process was complicated by the large number of legacy and exceptional constructs that needed to be maintained and secured.

A Shift in Mindset

The API management team embraced the idea that API configuration was an engineering process. It started viewing APIs as a composition of various product features, including API productization, security settings, network routing, and infrastructure services. All these features required inputs—some mandatory, some optional—that could be validated automatically in most cases. This shift in mindset enabled the reduction of the API management team workload.

First, the team started collaborating with product teams to identify the product features of each API and the corresponding parameters required based on security guidelines. Additionally, the team created a knowledge base that translated the product features into the actual deployment descriptor needed for the CI/CD automation tooling.



Product features are the important characteristics that describe how a given API is expected to operate. These features were split into three large categories:

- API functional and security definitions: Air France-KLM APIs requiring specific authentication and security settings. An example is "an API carrying confidential data intended for machine-to-machine exchanges."
- **API routing:** APIs that used an access path from TIBCO Cloud API Management to back-end services. Because the Air France-KLM network follows the defense-in-depth principle, a connection to the back-end is possible via a chain of security devices. An example could be "an access path for public-facing, high-volume services with added guards against site-scraping misuse."
- Back-end expectations and infrastructure service dependencies: APIs with dependencies and developer expectations for a back-end service such as "generates standard Air France-KLM HATEOAS links," "supports users being authenticated via strong B2E authentication," or "incompatible with standard B2E standard authentication."

To describe features, the API management team moved away from the challenging XML-based deployment descriptors and developed a lightweight, robust domain-specific language (DSL) based on a popular <u>YAML</u> data serialization language. This DSL emphasizes expression succinctness, as seen by the internal nickname DRY or, "don't repeat yourself." Product teams easily understood it after just a short introduction to the enterprise API landscape. Consider this practical example. the DRY format requires 44 lines and 643 characters, while the XML descriptor requires 152 lines and nearly 10 kilobytes. In terms of character volume, it's a compression of more than 10 times!

DRY FORMAT	XML DEPLOYMENT DESCRIPTOR
package type: rest name: Aircon Ground domain: passenger	xml version="1.0" encoding="UTF-8" standalone="yes"?
	<udm.deploymentpackage application="mashery_conf/
Aircon-ground REST mash" version="ITE1.T1.0.0-
ac95b0c-RS-0.9b2"></udm.deploymentpackage>
version: 1.0.0 path: /aircon-ground	<pre><orchestrator>sequential-by-container</orchestrator></pre>
	<deployables></deployables>
notify admin emails: - someone@klm.com	<mash.servicedeployable name="aircon-
ground-REST_service"></mash.servicedeployable>
<pre>responsible emails: - someone@klm.com expose: - asset spec: version: V01 type: rest path: "" exposure policy: http verbs: - get - post - put</pre>	<pre><servicename>ITE1 Aircon Ground REST</servicename></pre>
	<pre><description>DRY-made on 28 11 2021 using rule set RS-0.9b2. XL Deploy application mashery_conf/Aircon-ground REST_ mash</description></pre>
	<pre><version>ITE1.T1.0.0-ac95b0c-RS- 0.9b2</version></pre>
	<pre> <qpslimitoverall>0</qpslimitoverall></pre>
	<rfc3986encode>false<!--<br-->rfc3986Encode></rfc3986encode>
	<endpoints></endpoints>
- delete	<pre></pre>
data center: ams platform: docker data center segment: kl_whz	72 lines of XML code
environment: itel	22 lines of XML code
<pre>provides from: "/aircon-ground/": assets: "/": asset name: Aircon Ground asset versions:</pre>	<mash.packagedeployable name="aircon-
ground-REST_Package"></mash.packagedeployable>
	15 lines of XML code
	<plans></plans>
	12 lines of XML code
	<pre><mash.plandeployable name="aircon-ground-REST_Package/Default"></mash.plandeployable></pre>
	<planname>Default</planname>

Compared to XML, the DRY format also requires considerably less typing and proofreading. The main benefit of adopting this approach was codifying the decisions and achieving decision consistency among team members. In the example above, when attempting to deploy this package, the deployer will see the following critical warning:

- REST API via SaaS without Strong Authentication enabled > Reason: Cross-check required to confirm input correctness
- > Response:
 - | Verify that the security exception is duly granted for this API.
 - \mid A security exception MUST NOT be used to suppress error code LM004.
 - | In case the requester just tries to work around CISO-mandated
 - | constraints, the API **MUST NOT** be deployed.
- $\-$ Set by rule common > Raise cross-check if REST service is requesting exposure via SaaS without Strong REST Auth with a security exception

^{--+-- 1} cross-check as follows: -----+ [LM005] Endpoint aircon-ground-v01 is granted security exception to expose

Deployment Time Reduced, Potential Errors Removed

How much time, effort, and financial cost did the Air France-KLM API management team save when it implemented these measures? The team quadrupled throughput with fewer people!

- The effort to set up a typical API deployment went from several hours to just minutes.
- The effort to set up more complex cases went from a week to just hours.
- The team's remaining three specialists can manage up to 30 API deployment projects at a time.

Most importantly, this gives Air France-KLM the agility to respond more quickly to changing business requirements by releasing new digital services to market faster.

Not only has efficiency increased, but the new process continuously removes chances to make an inadvertent error. The current knowledge base features 500-plus rules supported by 300-plus regression test scenarios. Whenever a team member discovers a new edge case, rules and regression tests are added.

Additionally, the DSL format streamlines employee on-boarding by simplifying the process and providing learning material for future team members. The company's internal documentation offers several solution templates that illustrate which product features to select to build APIs. Product team architects can easily use DSL with complementary tooling to investigate and visualize the effects of various API publishing configurations before deploying anything. Given that these configurations can now be built in minutes, the architect gains the ability to evaluate and choose the most optimal deployment strategy for the product or release.

Additionally, for the product teams with high devOps maturity, Air France-KLM has started offering the possibility to send pull requests to directly propose desired changes. In an ideal case, the API management team will only need to review and approve these requests, reducing the API management team's workload even further.

By implementing and streamlining automated decision-making in its daily work, Air France-KLM has delivered on its strategic objective: Achieve first-time-right API deployment even in the midst of post-COVID resource constraints.



Global Headquarters 3307 Hillview Avenue Palo Alto, CA 94304 +1 650-846-1000 TEL +1 800-420-8450 +1 650-846-1005 FAX www.tibco.com

By implementing

and streamlining

decision-making

in its daily work,

Air France-KLM

has delivered

on its strategic

objective: Achieve

first-time-right API

deployment even in

the midst of post-

COVID resource

constraints.

automated

TIBCO Software Inc. unlocks the potential of real-time data for making faster, smarter decisions. Our Connected Intelligence platform seamlessly connects any application or data source; intelligently unifies data for greater access, trust, and control; and confidently predicts outcomes in real time and at scale. Learn how solutions to our customers' most critical business challenges are made possible by TIBCO at www.tibco.com.

©2022, TIBCO Software Inc. All rights reserved. TIBCO, the TIBCO logo, and TIBCO Cloud are trademarks or registered trademarks of TIBCO Software Inc. or its subsidiaries in the United States and/or other countries. All other product and company names and marks in this document are the property of their respective owners and mentioned for identification purposes only.

25Jan2022