Architecture & System Integration

Ryanair Labs

By Emil Gustafsson

LL

# Table of Contents

[Table of Contents ii](#_Toc22906266)

[Introduction 1](#_Toc22906267)

[1. Information Management and Integration 2](#_Toc22906268)

[2. Application and Data Storage Infrastructure Design 2](#_Toc22906269)

[3. Application and Service Integration 3](#_Toc22906270)

[4. RESTful Web Services 4](#_Toc22906271)

[5. Mashups 5](#_Toc22906272)

[6. Assumptions 5](#_Toc22906273)

[7. Conclusions 5](#_Toc22906274)

LL

# Introduction

In this report we will discuss three topics covering to the architecture, information and system integrations relating to Ryanair.

We will provide and explain our mashup demo application that allows for planning of aircraft flight paths, and the saving and loading of these routes.

# Information Management and Integration

Enterprise Information Integration means to provide uniform access to multiple autonomous, distributed data sources that is using different formats.

There are three types of Information Integration Technologies relevant for Ryanair.

**Data Federation** is a data integration technology that aggregates data from multiple disparate sources into a virtual database and allows for the database to be queried real-time. This is then used by business intelligence, reporting or analysis applications.

This federated data view doesn’t contain any real data of its own, but rather metadata about the data or the location of the data. The actual data stays in the source place.

Some pros of data federation are that it provides a current in real-time view of transactional data that’s being stored in multiple places. Only the requested data is returned, and it reduces data storages and data transfers.

Some cons and limitations of data federation is that it still queries against data sources, and therefore only contains as much data. And if the source data is archived off source it’s no longer available from the federation standpoint.

**Data Consolidation** is a technology means to collect data from multiple different sources (be it databases, internet or cloud) and centralize this into a central storage like a Data Warehouse.

This can be good, because having your data in one place makes it easier and more accessible. But it also offers risk, as a single point of failure if something goes wrong.

**Data Propagation** technologies means to distribute data being stored in one or more source data warehouses into one or more data sources, and letting data propagate over different systems in time. It gives quick access locally, and propagates to the different departments and locations over time.

Ryanair can use data federation to analyze their data and gain more understanding and insights into its customers and their habits, and tailor services and make adaptions to their needs.

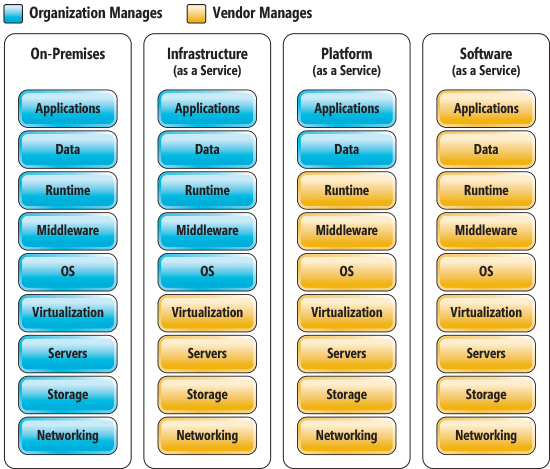
Ryanair can use data consolidation to collect the data from their many disparate systems and collect them in data warehouses.

Data propagation is good in the case of Ryanair, who has its teams, departments and systems spread across a big geographical distance.

# Application and Data Storage Infrastructure Design

Cloud Infrastructure refers to all the components, hardware or software, which are needed to support the computing requirements of a chosen cloud computing model. In cloud computing, virtualized resources are hosted by service providers to customers over network or internet.

There are different types of Cloud Servicing Models, which offers differing levels of resources. These different models provides different levels of resources, and therefore requires certain layers to be provided by the customer.



Depending on the system, data or processes that are being outsourced in the chosen Cloud Servicing Models, different pros and cons must be considered. For example, a department having an application outsourced as Software as a Service will be locked into the functionality provided by the cloud provider. While under a Platform as a Service the application can be managed by the Ryanair IT department, which can make updates or modifications to the application.

It’s good business for enterprises to move their computing power to the cloud and utilise an elastic computing design. Because this means an on-demand computing supply, where the resources are expanded or shrunken to meet the demand of the enterprise.

Ryanair have some traffic peaks during some holidays. Hosting its own infrastructure and resources to meet these high points, that only comes a couple of times a year, might be a bad equation. That infrastructure will not be utilised to the maximum the rest of the year.

Moving computing power to the cloud makes excellent financial sense. However, Ryanair must as all enterprises be mindful of the processes and the data it outsources. The outsourcing must follow laws and regulations and the enterprises IT Governance.

# Application and Service Integration

Enterprise Application Integration, meaning the process of sharing data and business processes among connected applications and data sources within the enterprise.

To integrate all the different application and services, and the data transferred between them, there are three main technologies utilised today.

The first one of these is the Direct Application to Application approach, also known as the “Spaghetti bowl”. It looks like a mesh network, with individually designed APIs for every connection. Meaning exponential number of connections with the increasing number of applications. Quickly making the system more complex and harder to maintain. Therefore, we don’t recommend this approach for Ryanair.

Ryanair could adopt the Hub-and-spoke integration, which means a centralised hub that handles the information exchange and transformation. We do not recommend this however, because it’s not easily scalable and means a single-point-of-failure if it stops working.

We instead recommend Ryanair to adopt the Enterprise Service Bus (ESB) approach, which is an open standard-based asynchronous messaging middleware solution. It provides secure interoperability between the enterprise applications. The basis of ESB is Service-Oriented Architecture, which is the architecture Ryanair has chosen on our recommendation. And because ESB, unlike Hub-and-spoke, can consist of and be distributed over multiple ESB services, it does not present a risk of single point of failures.

# RESTful Web Services

Web Services are codes or applications meant to be interacted with by other machines or applications, unlike Web Applications where a person is the consumer.

RESTful (Representational State Transfer) web services are, unlike the common web applications, based on virtual folders. Where classical web application hierarchy is based on folders and the files in it, RESTful web services looks at the URL provided and directs the requester to a virtual folder structure.

Take, for example, this URL: <http://example.com/login?username=peter&password=secret>

The RESTful web service directs the requester to the virtual “login” folder and passes data as arguments in “username” and “password”.

Our microservice.py for the Google Maps Mashup consists of two virtual folders, one for each of the two main functions. One is “getRoute”, which will return a saved XML-object specified by name. The other one is “saveRoute”, which receives an XML-object and saves it into a file.

Running the code for the micro service is simple. All you have to have installed is Python 3.X and the bottle-package for Python. With these installed, just start the ***microservice.py*** by double clicking on it. If using an IDE like PyCharm, load the file into the IDE and press “Run”.

Now the service is running in the background, and should log all requests received from browsers in the console.

Open the ***plan\_route\_map.html*** file in your browser, and the interface of our mashup service is displayed. At the top of the webpage are the menu, which contains an input text-field. This field is used to enter the name of the route you want to save or load.

By clicking the mouse on the map, waypoints will be placed with a polyline between them. This represents the flight route you’re planning. When done, press the “Save Route” button and the route will be saved using the name entered into the text field to the left.

# Mashups

A mashup is a small and lightweight web application that combines data or capabilities from multiple sources to deliver a new functionality.

In our Mashup Demo we have combined data from Google Maps in the form of a map, with routes consisting of coordinates entered through a web page into our own micro service.

A very handy functionality in creating mashups is Ajax, meaning asynchronous JavaScript & XML. It allows for web applications to web send request without the user having to update the web page, by sending requests via JavaScript. This is done in our demo when saving a route, or loading one.

Usually, web requests from domains outside the domain from which the first resource was shared isn’t allowed for security reasons. For example, Ajax requests are by default not allowed according to the same-origin security policy.

To allow for cross-domain request, a mechanism called Cross-Origin Resource Sharing (CORS) was created. CORS defines how a browser and the web server can interact to determine whether it is safe to allow cross-origin requests. This is done by adding an “Access-Control-Allow-Origin” in the header of the web requests.

# Assumptions

* I have assumed that the map should be cleaned of waypoints when loading an already saved map. So the user should not have to clear it manually before loading a route.

# Conclusions

We have concluded that Ryanair should implement an Enterprise Serial Bus, and not the spaghetti bowl or hub-and-spoke, for the Enterprise Application Integration.

We explained the different information integration techniques, and how these can be beneficial for Ryanair.

We talked about different cloud options and how these should be considered before implementing.

We explained mashup and RESTfull web services, and how these relate to our demo mashup.