Case Study

AnyAirline GetAways

# Enterprise Overview

The established, global airline AnyAirline is launching a new line of business called **GetAways**, nominally to revitalize relationships with existing customers. The target market is middle class travelers who infrequently travel for leisure. GetAways are packaged holidays and other travel bundles that are designed specifically for each customer. They view their competitors to be other online travel booking companies such as Expedia, HipMunk, and Travelocity, as well as newer emerging startups who have already embraced **digital** **transformation** to focus on **customized** **travel** **experiences** based on intimate knowledge of the traveler's interests and social networks driven by innovative big-data solutions.

The GetAways line of business does is not completely independent, and depends on AnyAirline shared services which include marketing, legal, IT, HR, and facilities. About a year prior to the GetAways debut, AnyAirline adopted a three year strategic plan, in which a major component is the Be There initiative. Be There includes diverse but interrelated initiatives designed to Show That We Care, Earn Trust, and Create Experiences.

GetAways is requesting support from marketing to manage its crucial first campaign. It happens that marketing department embarked on a nine month project Knowing Every Traveler about a month ago. The project is intended to enrich AnyAirline’s relationship with individual customers by first perceiving, then understanding, and ultimately redesigning the narrative of interactions each person has with the brand. Under advisement by IT, Marketing has acquired a Customer Relationship Management platform (salesforce.com) to enable this initiative.

## Integration Team

A new **GetAways architect** is being added to the **AnyAirline enterprise integration team**, which falls under the **Data Services** department. Data Services is responsible for data **warehousing**, **integration**, and **reporting** services for the rest of the enterprise. After a reorganization two years ago, enterprise integration falls under the authority of the **Marketing department**, while responsibility for infrastructure and database administration remains in **central IT**.

The **Director of Integration Architecture** spearheaded the **adoption of Mule** and the **Anypoint Platform** just over a year ago. There are currently two Mule applications in production running on CloudHub. In anticipation of an upcoming project, a proof of concept integration is being developed to demonstrate connectivity between CloudHub and the AnyAirline corporate datacenter.

# The Plan

Currently, all customer representatives **search for flights and related goods and services** for customers using an **enterprise legacy system**. Customer representatives can also book travel and make reservations for related goods and services. There is also a web site that customers can use for self-service, and to view and manage already booked reservations.

A **legacy system** already allows the customer support team to create some new **travel offerings**, and **run marketing campaigns**. However, as part of the GetAways initiative, AnyAirline has selected a Salesforce.com **CRM** **system** as its strategic **customer management tool**. In the first phase of the project, the CRM system is going to manage only customer data for **vacation travel bundle** marketing campaigns.

Customer login information and group membership is stored in a corporate Active Directory server which is used to support single sign-on.

Currently, AnyAirline (the parent organization) wants to maintain **customer information** in **both** the **legacy enterprise system** and the **CRM system**.

Therefore, it is important for **GetAways** **to synchronize customer information from the legacy enterprise system to the CRM system, and vice-versa**.

The Director of Integration Architecture has made it a key performance objective that we eliminate point-to-point integration. Instead, this solution should design System APIs that functionally decompose and abstract away point-to-point integrations between different physical systems into more reusable and composable parts.

The GetAways crucial first marketing campaign is planned to launch in 3 weeks, and it relies on analytics and targeting capabilities of the CRM platform. The marketing team needs 5 quarters of customer profiles, purchase, and cancellation history to be migrated to the new platform quickly, and will need about three days to work with the data for the campaign to succeed.

# General Requirements

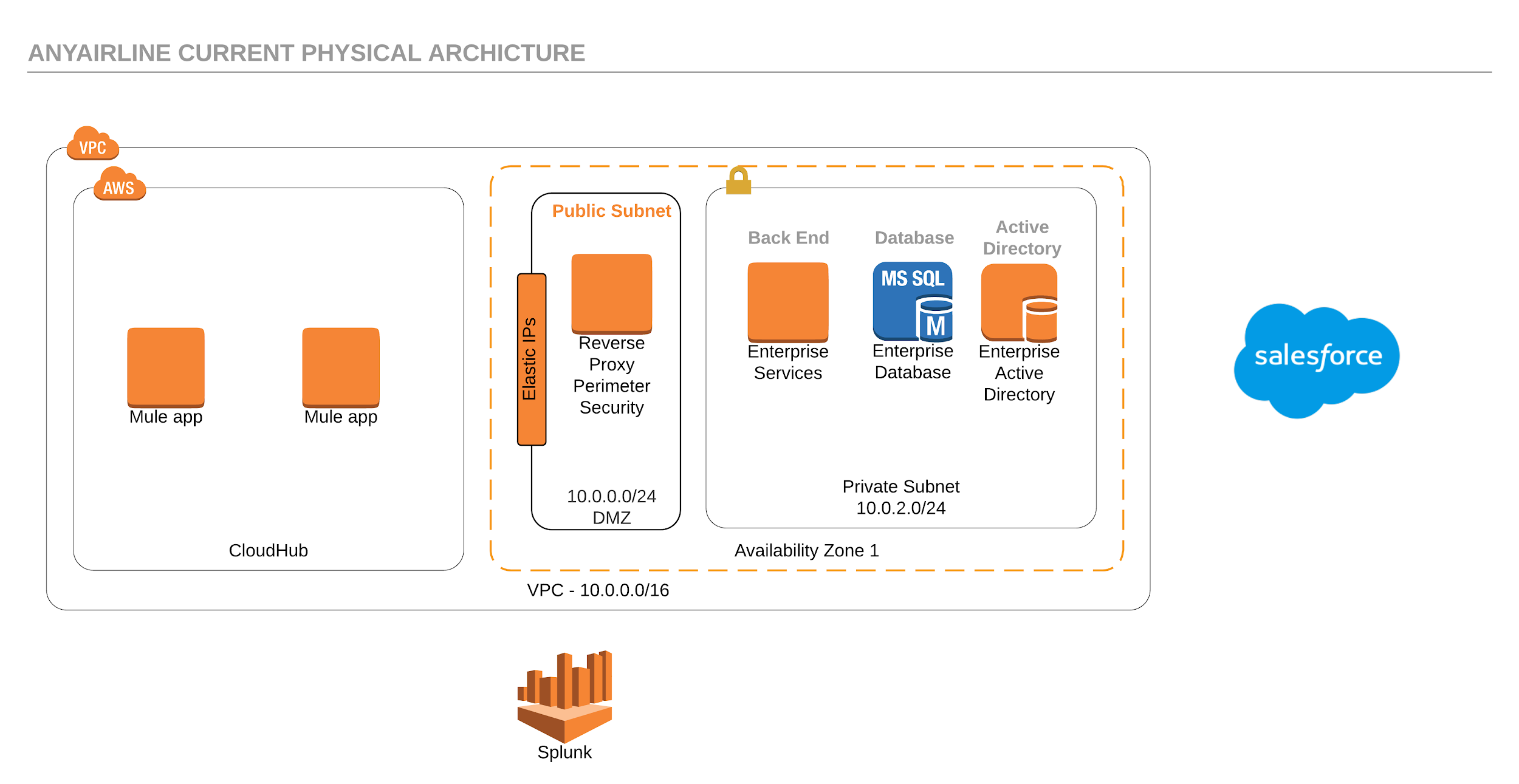
Preexistent systems with which some integration is required, which are out-of-scope for this phase of solution design and implementation:

* Flights Management system
  + Accessible via SOAP web services over mutually authenticated HTTPS.
  + Deployed on-premises in the AnyAirline data center.
* Passenger Data system
  + In-house legacy PostgreSQL database to be accessed directly.
  + Deployed on-premises in the AnyAirline data center.
* Salesforce CRM
  + Recently introduced.

# 

# Physical View of the GetAways Organization Systems

This diagram summarizes the current physical view of the enterprise systems involved in the integration project.



# Functional Requirements

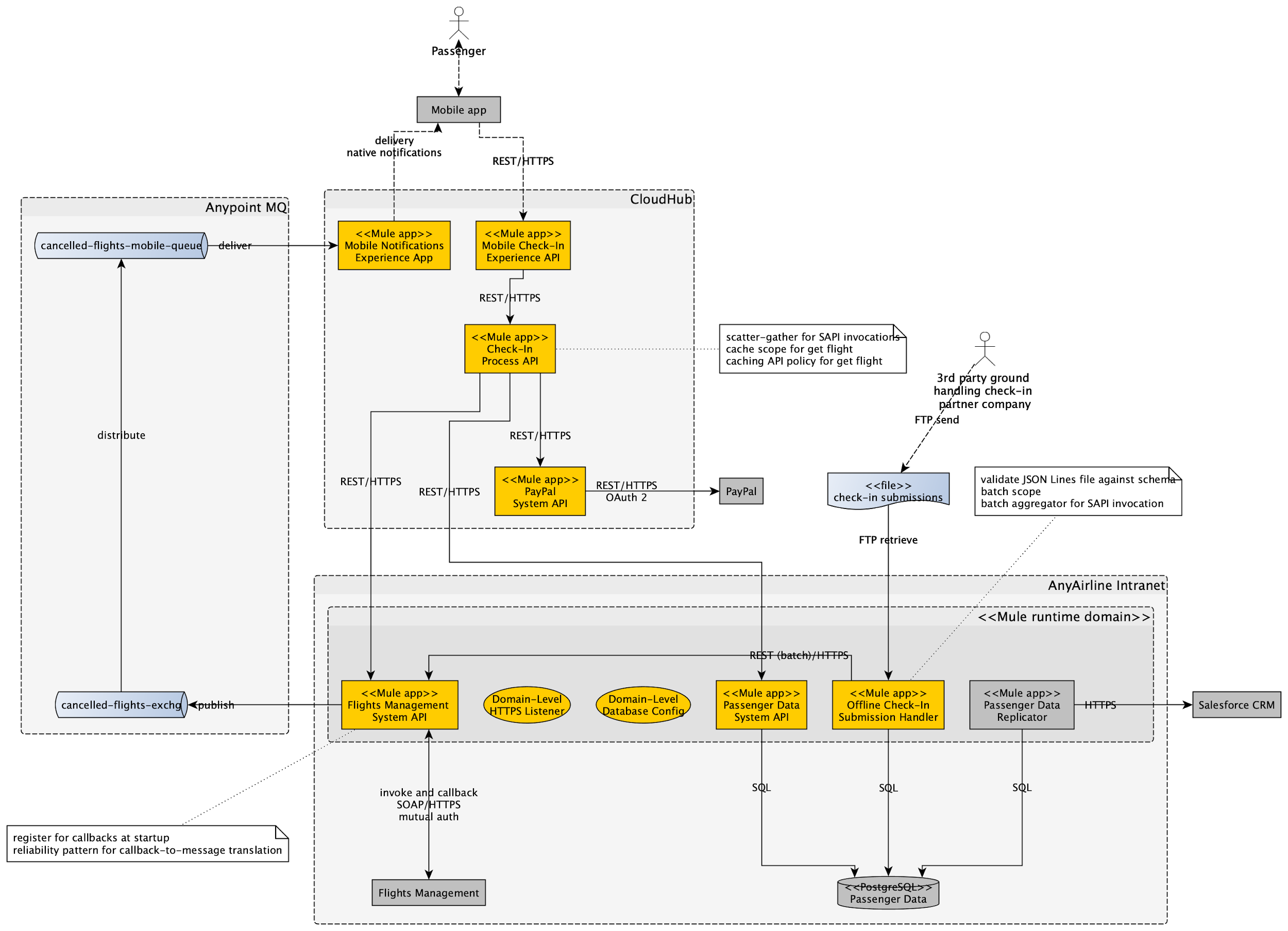
### Actors

* Traveler
  + A customer with a booked travel reservation.
  + AnyAirline uses the term Traveler to refer to a customer with a booked flight on AnyAirline
* Marketing Campaign Manager
  + Responsible for improving customer loyalty and attracting a new market segment
* Travel Agents
  + Support passengers to book travel

# High-Level Solution Architecture

In addition to the GetAways launch, AnyAirline has several ongoing initiatives across the organization. The MuleSoft C4E has created a high-level architecture to show how these various ongoing solutions tie together.

This free-form diagram gives an overview of the components under design and development (shown in yellow and blue) and the preexisting systems to integrate with (or otherwise out-of-scope in this project phase; shown in grey). Comments allude to implementation features of some of the application components, some of which will be addressed in more detail later.



# 

# Phase 1: Loading and two-way database synchronization

GetAways plan is to initialize the integration effort by migrating all existing customer contacts from the AnyAirline's legacy system to the GetAways Salesforce system first.

Afterwards there needs to be a set of applications that will maintain contact synchronization between the Salesforce system and AnyAirline's legacy enterprise system. This means that if:

1. AnyAirline has an addition, deletion, or update on a contact record in its legacy enterprise system, the record will get updated accordingly within the GetAways Salesforce system.
2. The GetAways Salesforce system has an addition, deletion, or update on a contact record, the AnyAirline legacy enterprise system will get updated accordingly.

The Synchronization application will interact with two System APIs (One for AnyAirline's Legacy Enterprise system and one for the GetAways Salesforce Account Objects) that will handle the CRUD operations for each system. These System APIs will need to be built first before the Synchronization application can be built.

## Functional Requirements for the Data Loading application

The data loading application must retrieve all customer information and efficiently load the information into the GetAways Salesforce Accounts database.

### The Existing Legacy Enterprise Database System

AnyAirline has a legacy database system that is used as an **enterprise system** (monolith system). This enterprise system is not scalable, so AnyAirline wants GetAways solution to throttle requests to this enterprise system. The enterprise system provides a SOAP service for CRUD operations to the database secured by two-way SSL communication with Basic Authentication.

The customer information contains **personally identifiable information** (PII) data such as social security numbers, pin numbers, and addresses. The PII fields are encrypted in transit from the SOAP service.

Therefore the data loading application must:

1. Securely connect to the SOAP service.
2. Throttle requests to the SOAP service.
3. Load data to Salesforce efficiently.

The PII data will stay encrypted on the GetAways Salesforce system and it will be the GetAways Salesforce system's developers' jobs to build applications to properly decrypt and encrypt information entered into the GetAways Salesforce system.

## Functional Requirements for the two System APIs

After the data has been loaded into the GetAways Salesforce system, the next step is to build System-level APIs that will be used by the Synchronization application to maintain synchronization across the two database systems.

### The AnyAirline Enterprise Database System API

The SOAP service provides methods for CRUD operations to the AnyAirline enterprise database. Therefore the system API will act as a wrapper to the SOAP service to execute the CRUD operations. This API has been designed to be reused by both GetAways and other AnyAirline LOBs.

The SOAP service returns a success or fail message if a CRUD operation is successful or not.

The System API then logs the success or failure of the CRUD operation to an external logging service.

If the operation is a failure, the previous information for the target account will be preserved on the enterprise database.

The API is secured by a two-way SSL with a Client-ID and Client-Password authentication set as an API-policy in the Anypoint platform.

### The GetAways Salesforce Account System API

The GetAways Salesforce system will expose account operations with a system API. This system API will expose a REST services to provide CRUD operations to the Salesforce database.

The System API will then log the success or failure of the CRUD operation to an external logging service.

The API is secured by a two-way SSL with a Client-ID and Client-Password authentication set as an API-policy in the Anypoint platform.

## Functional Requirements for the Synchronization Application

A synchronization application will listen for changes in either database system. This application belongs in the Process API layer.

This process API will use Salesforce webhooks that allow an application to listen to event changes in the Salesforce system.

Your IT department has created a webhook to push notifications to a message-broker whenever an account has changed within the enterprise database system.

Therefore, the synchronization application must:

1. Take CRUD notifications from the GetAways Salesforce system and update the account within the AnyAirline enterprise database.
2. Take CRUD notifications from the AnyAirline enterprise database and update the account within the GetAways Salesforce system.

# Phase 2: Loading customer financial data into a more modern database

AnyAirline currently stores customer loyalty and other financial data related to each customer in an old mainframe server. AnyAirline wants its IT department to stop developing on the mainframe server due to the high technical debt the mainframe server imposes, so would like GetAways to avoid using the old mainframe system.

The first department that will move away from requiring access to the mainframe server is marketing. Therefore new GetAways Mule applications must pull all loyalty program and financial data into a new database for the past 5 years.

Because of financial regulations, all financial data cannot cross over to any system outside its on-premise location. Therefore the architecture team has decided to run the new database and Mule runtime in an on-premises (customer-hosted) location.

### Functional Requirements for the Marketing Upload application

Development to access the mainframe server is cost prohibitive, and the mainframe server team has therefore decided to export the data into flat files to an SFTP server located in the on-premises location. The mainframe server will then export new data to the SFTP server on a daily basis.

Therefore the application must:

1. Load the flat files from an SFTP server
2. Clean up the fields and transform the nested data into a denormalized structure
3. Load the data into a modern database using Transactions
4. Log any errors if the data loading fails for any record

# Phase 3: Using Machine Learning for marketing

The next phase in GetAways API-led development is to populate a marketing database with a prediction of how likely a customer is to respond to marketing campaigns for travel to a specific region or with specific partners.

There needs to be a system that will periodically evaluate every active customer in the GetAways Salesforce system and score the customer based on their:

1. Travel and loyalty history and usage
2. Demographics
3. Past responses to past marketing initiatives

The evaluation score will sent to a marketing database. Marketing will then use the score to help the customer support team run marketing campaigns.

The Mule application will retrieve loyalty program data and financial information from the modern database that was created in Phase 2. Customer information on demographics and past marketing activity is located in a Marketing API that Marketing will provide. The Marketing API is secured through Two-Way SSL using OAuth authentication.

Because of financial regulations and AnyAirline's data policies, all financial data cannot cross over to any system outside its on-premise location. The architect has decided to run the Mule runtime in a customer-hosted location.

Therefore the design of the scoring application will involve running several Mule applications on the same Mule runtime running in the customer-hosted location.

## Functional Requirements for the Data Pipeline Application

The first step is to create a Mule application to gather all necessary data from several systems and prepare the data to be sent off to the machine learning application for scoring.

The Data Pipeline application will schedule itself to:

1. Retrieve loyalty program and financial data from the modern database located on-premises
2. Retrieve demographics and past marketing activity information from the Marketing API in the cloud
3. Clean, transform, and combine all data into a format that the machine learning model expects
4. Run in a non-default Mule domain.
5. Send the data to a messaging queue (such as JMS or VM) to allow another Mule application to pick up and score.

## Functional Requirements for the Machine Learning Application

**GetAways'** Data Science team has provided the machine learning model to be used for scoring. The model is serialized as a Java object and needs to be deserialized using the

[Weka Machine Learning library](https://www.cs.waikato.ac.nz/ml/index.html) within the Mule application.

The machine learning application will:

1. Run in the same domain as the Data Pipeline application
2. Process data from a messaging queue (such as VM) that the Data Pipeline will push data towards
3. Score the data
4. Push the score to a database using Transactions
5. Log any data loading and scoring errors to an external logging service