CRM Pilot Enablement

System Architecture Document

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <31/Jan/19> | 1.0 | Initial walkthrough of exercises | Ryan Hoegg |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[Introduction](#_kdeuoxbbrzc)

[1.1 Purpose](#_nlfs80ohxtfu)

[1.2 Scope](#_g6fbbg48ocse)

[1.3 Definitions, Acronyms, and Abbreviations](#_glape1ch94ib)

[1.4 References](#_xbn7coju2x3a)

[1.5 Overview](#_i69czmg510fv)

[1.6 Assumption](#_yomg3skotl37)

[2. Architectural Goals and Constraints](#_pv4x3rxomye2)

[2.1 Context](#_wopycba3k0s7)

[2.2 Current State](#_92xti5iiptie)

[2.3 Approach and Methodology](#_oezx0jpzu1zs)

[2.3.1 API Led Connectivity (Conceptual Approach)](#_g4o6m2irsjrm)

[2.3.1.1 System Layer](#_9yjhz9oqaqnv)

[2.3.1.2 Process Layer](#_tv5766n4k5lh)

[2.3.1.3 Experience Layer](#_hlvvgd7yrteq)

[2.3.2 Mapping with <CUSTOMER> Methodologies](#_7fz02rogx0c6)

[3. Architectural Representation](#_7rsvucn5hbqq)

[4. Use-Case View](#_w5eldp45o4ia)

[4.1 Use-Case Realizations](#_gq6xg9g8c9m0)

[5. Logical View](#_dgwd70kvrt04)

[5.1 Overview](#_c6uwr48khkiu)

[5.2 Architecturally Significant Logical Flows](#_6jsp8ggsco4i)

[6. Deployment View](#_vv8kzzgdnab5)

[7. Process View (optional)](#_vpl9wpm3jg93)

[8. Implementation View (optional)](#_qgid5cvipkd0)

[8.1 Overview](#_7ker61pttsq2)

[8.2 Layers](#_z9z6dsl1x19s)

[9. Data View (optional)](#_lcb4ryxpjwh5)

[10. Integration Patterns](#_wvbskiea26d6)

[11. Software Development LifeCycle](#_kf0hss9gcaak)

[11.1 APIs LifeCycle](#_l9ukv7ws9d7a)

[11.2 Development principles / guidelines](#_rbdyv9gfu8g8)

[12. Size and Performance](#_bupw63jchcw9)

[12. Non Functional Requirements and Service Level Agreements](#_u79t8q3o9v29)

[12.1 Failure Points Analysis](#_oy0z1tqiqomq)

[12.2 Mechanisms](#_tgfsjakqggej)

[12.2.1 Specific Mechanisms](#_4znt2istkdfa)

[12.2.1.1 Testing Strategies](#_ga7tjl7xatrw)

[12.2.1.1 Performance Testing Plan](#_muy7b08a9uj4)

[12.2.1.3 Performance Testing Tools](#_1r9be3knrcy6)

[12.2.1.4 Performance Testing load scenarios](#_j3ngs6oc5f48)

[12.2.1.5 Dashboard of interest in Iterations](#_vlzhknfukl4s)

[12.2.1.6 Alerts/Notifications](#_bac0sqtacnf)

[12.2.1.7 Logging](#_qa8h928991hd)

[12.2.1.8 Error Handling](#_ag1frqr8sd7k)

[12.2.1.8.1 Validation Strategy](#_i6srl1c5mzot)

[12.2.1.8.2 Error Handling Strategy](#_6az20blw1ak4)

[12.2.1.8.3 Flow Level Exception Handling](#_8e2m7zgf2mx4)

[12.2.1.8.4 Application Level Exception Handling](#_xorko0dczowo)

[12.2.1.8.5 Global Exception Handling](#_amb9mfltrn3i)

[12.2.1.8.6 APIKit Exception Handling](#_lbx5s3fevzic)

[12.2.1.8.7 HTTP Status Code](#_43kabhkciece)

[12.2.1.9 Failure recovery strategies](#_y7pyjwqizsgy)

[12.2.1.9.1 Reconnection Strategies for connectors](#_cv0w6nir64xv)

[12.2.1.9.2 Retry strategies for flows](#_cv0w6nir64xv)

[12.2.1.9. 3 Redelivery strategies for flows](#_cv0w6nir64xv)

# System Integration Architecture Document

# **Introduction**

The “CRM Pilot” refers to an effort underway within the AnyAirline marketing department, as part of the *Be There* initiative. *Be There* is a key part of AnyAirline three year Strategic Plan, which includes diverse but interrelated initiatives designed to *Show That We Care, Earn Trust,* and *Create Experiences.*

An important part of this initiative is a nine month Marketing initiative *Knowing Every Traveler* intended to enrich our 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.

There is a marketing campaign planned to launch in 3 weeks that 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.

## **1.1 Purpose**

This document’s primary purpose is to enable the AnyAirline marketing department to accomplish a successful CRM Pilot, by determining an optimal integration architecture for the solution. It is meant to include all relevant design decisions, constraints, and success criteria for the solution, and provide sufficient guidance for the successful delivery of the solution and its ongoing operation.

We also intend this document to serve as a reference for other integration efforts, through AnyAirline C4E initiative *(intranet link)* Thus, while there is normally little need to make updates once the solution has been delivered, we plan to maintain this document’s structure and organization on an ongoing basis. We also plan to include much more detail in some sections than strictly needed for the Pilot, in order to serve as reference for other initiatives.

## **1.2 Scope**

This solution is intended to address integration requirements relevant to the introduction of the salesforce.com CRM system at AnyAirline. These include the bidirectional synchronization of customer data between CRM and the legacy customer system,, and the one time load of customer, purchase, and cancellation data needed for the GetAways marketing campaign.

This solution makes no substantial changes to the physical architecture in place (notably, Mulesoft and the legacy database).

## **1.3 Definitions, Acronyms, and Abbreviations**

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the **System Integration Architecture Document**. This information may be provided by reference to the project’s Glossary.]

|  |  |
| --- | --- |
| **Term** | **Definition** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## **1.4 References**

[This subsection provides a complete list of all documents referenced elsewhere in the **System Integration Architecture Document**. Identify each document by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]

|  |  |  |
| --- | --- | --- |
| **Author** | **Document Name** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## **1.5 Overview**

This document is intended to serve as the reference for the CRM Pilot solution architecture. That is, the set of requirements, design decisions, constraints, and technical specifications chosen and recognized by the AnyAirline enterprise integration team. To help readers and authors more efficiently locate a particular topic, this document is organized according to the 4+1 model in common use, following the format found in the AnyAirline Template: System Integration Architecture.

## **1.6 Assumptions**

[List of assumptions that were made while writing this **System Integration Architecture Document**.]

# **2. Architectural Goals and Constraints**

## **2.1 Context**

The solution must ensure the availability of useful data in the CRM system in time for the marketing campaign to be designed and launched. Because of the urgency of this goal, the solution prioritizes the quick delivery of a satisfactory solution for this one use case.

Expectations:

* <Define Cloud/OnPrem/Hybrid Architecture>
* <Setup environments (sandbox and production)>
* <Setup Identity Management>
* <Review of n use cases>
* <Core Framework definition>
* <C4E>

## **2.2 Current State**

* <CUSTOMER experience developing with Mule.>
* <CUSTOMER subscription (OnPrem | CloudHub | Hybrid, API Management + Analytics?, VPC, etc)>
* <# of expected use cases>
* <Source code tools used, GitHub>
* <Documentation Tools used>
* <CI/CD Tools used>
* <Stakeholders and roles>
* <Mule version and tools to be used>

# **3. Architectural Representation**

[This section describes what software architecture is for the current system, and how it is represented. Of the **Use-Case**, **Logical**, **Process**, **Deployment**, and **Implementation Views**, it enumerates the views that are necessary, and for each view, explains what types of model elements it contains.]

# 

# **4. Use-Case View**

This solution initially includes only one use case:

* As a marketing campaign manager, I want Provide the customer data needed for the marketing campaign within the new CRM system.
  + Customer Profile
  + Purchase History
  + Cancellation History

## **4.1 Use-Case Realizations**

# **5. Logical View**

[This section describes the architecturally significant parts of the design model, such as sequence diagrams. You should introduce architecturally significant sequence diagrams are typically associated with use case realizations in the Logical View of the system under development.]

## **5.1 Overview**

[This subsection describes the overall design model in terms of message exchange between different processes or objects that live simultaneously.]

## **5.2 Architecturally Significant Logical Flows**

[For each significant processes, include a subsection with details about flows with exchange of messages between components.]

# **6. Deployment View**

[This section describes one or more physical network (hardware) configurations on which the software is deployed and run. It is a view of the Deployment Model. At a minimum for each configuration it should indicate the physical nodes (computers, CPUs, vCores) that execute the software and their interconnections (bus, LAN, point-to-point, VPC and so on.) Also include a mapping of the processes of the **Process View** onto the physical nodes.]

The infrastructure is composed and divided by the following environments:

* Prod
* Sandbox

Example:

# **7. Process View (optional)**

[This section describes the system's decomposition into lightweight processes (single threads of control) and heavyweight processes (groupings of lightweight processes). Organize the section by groups of processes that communicate or interact. Describe the main modes of communication between processes, such as message passing, interrupts, and rendezvous.]

# **8. Implementation View (optional)**

[This section describes the overall structure of the implementation model, the decomposition of the software into layers and subsystems in the implementation model, and any architecturally significant components.]

## **8.1 Overview**

[This subsection names and defines the various layers and their contents, the rules that govern the inclusion to a given layer, and the boundaries between layers. Include a component diagram that shows the relations between layers. ]

## **8.2 Layers**

[For each layer, include a subsection with its name, an enumeration of the subsystems located in the layer, and a component diagram.]

# **9. Data View (optional)**

[A description of the persistent data storage perspective of the system. This section is optional if there is little or no persistent data, or the translation between the Design Model and the Data Model is trivial.]

# **10. Integration Patterns**

<Describe the patterns to be tackled>

Examples:

# **11. Software Development LifeCycle**

Source Code:

Staging

* Local Development
* Development environment
* Testing
* Intermediate stages (QA/UAT)
* Production

Testing

* Unit
* Integration
* Regression
* Performance
  + Load
  + Stress
  + Soak
* User Acceptance

## **11.1 APIs LifeCycle**

<links>

## **11.2 Development principles / guidelines**

* Mule Development Recommendations
* Naming Conventions

# **12. Size and Performance**

[A description of the major dimensioning characteristics of the software that impact the architecture, as well as the target performance constraints.]

# **12. Non Functional Requirements and Service Level Agreements**

[A description of how the software architecture contributes to all capabilities (other than functionality) of the system: extensibility, reliability, portability, and so on. If these characteristics have special significance, such as safety, security or privacy implications, they must be clearly delineated. The description of service level agreements for the system and as well as dependent system must be documented. ]

## **12.1 Failure Points Analysis**

Example:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Action | Failure (xx process) | | | |
| SFDC | Mule process | Message Broker | Backend app |
| SFDC | X | Notifications Replay manual process + 24 hrs restriction | Nothing | Nothing |
| Mule process | Reconnection Strategy | X | Strategy (retry + backup) | Nothing |
| Message Broker | Nothing | Nothing | X | Nothing |
| Backend app | Nothing | Nothing | Nothing | X |

## **12.2 Mechanisms**

### 12.2.1 Specific Mechanisms

#### 12.2.1.1 Testing Strategies

#### 12.2.1.1 Performance Testing Plan

#### 12.2.1.3 Performance Testing Tools

#### 12.2.1.4 Performance Testing load scenarios

#### 

#### 12.2.1.5 Dashboard of interest in Iterations

#### 12.2.1.6 Alerts/Notifications

#### 12.2.1.7 Logging

#### 12.2.1.8 Error Handling

##### 12.2.1.8.1 Validation Strategy

##### 12.2.1.8.2 Error Handling Strategy

##### 12.2.1.8.3 Flow Level Exception Handling

##### 12.2.1.8.4 Application Level Exception Handling

##### 12.2.1.8.5 Global Exception Handling

##### 12.2.1.8.6 APIKit Exception Handling

##### 12.2.1.8.7 HTTP Status Code

#### 12.2.1.9 Failure recovery strategies

##### 12.2.1.9.1 Reconnection Strategies for connectors

##### 

##### 12.2.1.9.2 Retry strategies for flows

##### 

##### 12.2.1.9. 3 Redelivery strategies for flows

# 